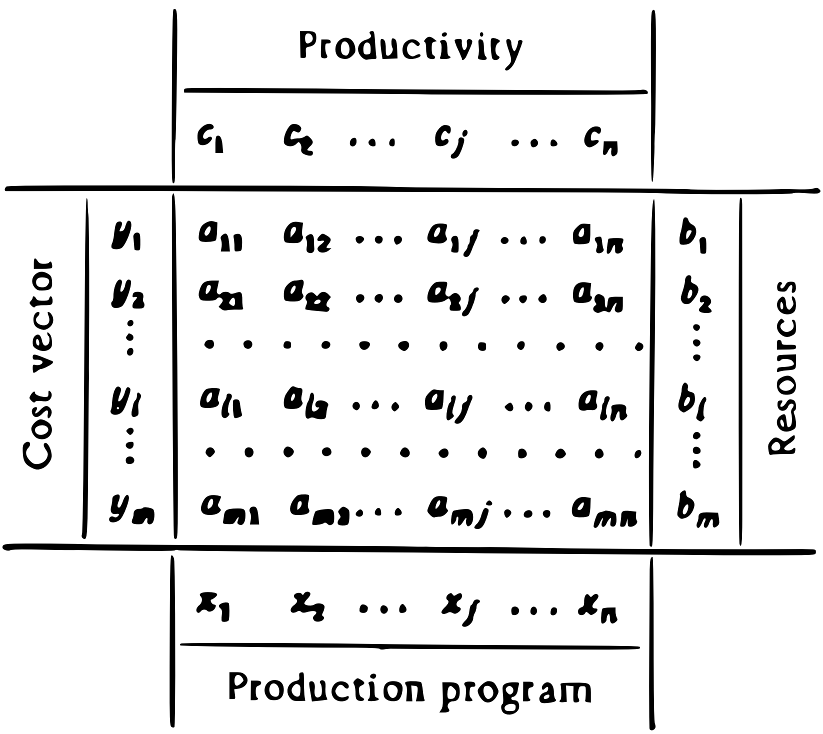
D.B. Yudin, E.G. Gol'shtein

LINEAR PROGRAMMING



D. B. YUDIN and E. G. GOL'SHTEIN

LINEAR PROGRAMMING

Translated from Russian by Z. Lerman

Israel Program for Scientific Translations
Jerusalem 1965

TABLE OF CONTENTS

ANNOTATION

Linear programming provides methods for computing the most practical solutions in production, operations, and supply planning and in the control of complex processes. The introduction of linear programming into practice materially cuts costs and time losses.

The present book presents in detail the mathematical theory of linear programming and computational methods yielding an exact solution over a finite number of steps.

The book is intended for engineers, economists, and applied mathematicians. It may also be used by university students in the mathematics, economics, and engineering economics departments.

PREFACE

Planning and control are among the most complex operations in industrial management, economic administration, and staff work at various levels. The actual procedure of planning and control determines the way to the final objective and essentially affects the quality of the decisions.

For a long time technological control and planning of economic development or strategic operations could be carried out only by individuals with suitable education, experience, and intuition. However, technological progress and present-day complexity of the structure of industry, distribution of labor and large-scale cooperation, all these have made the problem of planning and control highly complicated.

To arrive at a sound decision, an enormous volume of information must be collected and processed. Responsible decisions often involve the fate of many people and high material cost. Now it is no longer enough to indicate the way to the final objective. Of all the possible ways it is necessary to choose the most economic, which allows for the specific features of flow and development of the controlled process and which gives best results in the given circumstances.

Previous methods of decision making in planning and control problems, entailing expansion of the administrative body, may eventually lead to total loss of control in the present state of technology.

The advent of high-speed computers created a powerful precedent for complete automation of the numerous control problems. However, this possibility could be realized only after the development of suitable mathematical methods reducing the solution of planning and control problems to a succession of automatic operations based on the initial input data.

The new mathematical discipline serving as the theoretical basis of the solution of planning and control problems was called mathematical programming.

In stating a mathematical-programming problem, the purpose of the control and the restrictions to be imposed on the choice of the control parameters, as arising from the specific features of the control process, must be clearly indicated. In solving the problem, the system of parameters ensuring optimal (in the sense defined) quality of the control process within the framework of the stated restrictions is chosen.

The most developed branch of mathematical programming is linear programming — theory and methods of solution of conditional extremum problems, in which the quality function is a linear function of the control parameters, and the restraints are linear equalities or inequalities.

Works on linear-programming were first published by Soviet scientists (/61–63, 66/). In recent years various collections have been published in the Soviet Union (/96–99/) presenting the reader with original and translated works dealing with particular methods and various applications of

linear programming. However, all these articles, and even the highly rigorous and concise presentation of the principles of linear programming in Kantorovich's book /65/ can hardly be used as a guide in the systematic study of this mathematical discipline. The short monographs of Barsov /3/, Churnes, Cooper, and Henderson /118/, and Wayda /15/ cover only a rather narrow field of problems relating to the principles of linear programming. The works of Gabr /17/, Gerchuk /21/, Kreko /69/, Reinfeld and Vogel /88/ are intended for the unqualified reader interested mainly in economic applications of linear programming. Mathematical rigor and comprehensive analysis are sacrificed here in favor of readability.

Among the most comprehensive and also sufficiently rigorous modern reference books in linear programming we must mention the book /129/ and also the translation of the monograph /18/ by Gass. However, the attempt to give in one book general and special problems, theory and methods, and mathematical principles and applications made it impossible for the writer to devote due attention to each matter.

The aim of the authors of this book is to present the main aspect of linear programming — the theory and the finite methods — which is of interest to mathematicians, economists, and engineers, and to try, in this way, to achieve better mutual understanding of specialists taking part in the statement and the solution of planning and control problems. This book, in considerably greater detail than any other publication known to us, presents the general linear programming theory and gives a very thorough description of the finite methods and the corresponding algorithms. Much attention is devoted to various modifications of methods and algorithms which enable the structure and the special features of modern computers to be most efficiently utilized. Geometrical and economic interpretations of problems and methods of linear programming dealt with in detail in this book should be of considerable help in understanding more thoroughly the theory and also in developing an intuitive grasp of the principles of computational methods and algorithms.

The book contains some new results published for the first time. The selection and presentation of the material is intended to emphasize that linear programming is not an independent discipline, but a branch of mathematical programming. In particular, the finite methods of linear programming are classified so as to provide an analogous approach to methods of solution of nonlinear extremum problems.

The structure of chapters adopted in this book makes it possible for specialists of different mathematical background and with different interests to choose a sequence of study which seems best suited for their needs. For the reader's convenience, we add an Appendix giving the elements of linear algebra and the concepts of many-dimensional spaces, which are useful in the mathematical apparatus of linear programming. It should be emphasized that to grasp effectively the methods presented in the book the reader must have knowledge not only of the computational, but also of the theoretical aspects of linear programming.

In Chapter 1 the basic concepts of linear programming are presented, the relative place of this discipline among other branches of mathematical programming is indicated, and various methodological problems, which are of significance in the statement of problems and assimilation of the methods, are discussed.

In Chapter 2 the relationship between linear programming and the theory of convex polyhedral sets is established. The interpretation of the problem in terms of the theory of convex polyhedral sets makes it possible to substantiate various important results of linear programming and helps in assimilating the geometrical side of the problem and of methods of its solution.

Chapter 3 deals with the most important theoretical question of linear programming — the duality theory. The duality theory plays an essential role in the construction of linear-programming methods. In this chapter the relationship between the solution of the dual problem and the so-called decision multipliers — analogs of Lagrange multipliers — are established. The results of this chapter make it possible to devise a uniform approach to the different procedures used in the development of linear-programming methods.

Linear-programming methods, like the methods of linear algebra, are subdivided into finite and iterative. Previous experience in the solution of applied problems shows the finite methods to be, on the whole, more effective. Chapters 4 to 8 of the monograph present, in detail, the theory and the computational algorithms of the basic finite methods of linear programming. When describing the methods the authors paid attention both to matters of principle, which are of interest in the further development of the methods, and to special features of the computational schemes, according to which the method and algorithm for the solution of any particular problem are chosen.

The methods and the algorithms are described in application to the general linear-programming problem given in the so-called canonical form with the variables bounded on one and two sides. In the concluding chapter a procedure is indicated for the construction of computational schemes for linear-programming problems written in different forms. This chapter also gives some modifications of the finite methods ensuring more effective utilization of the working memory of the digital computer.

The classification of the methods according to the various criteria given in Chapter 8 leads to the conclusion that the methods discussed in this monograph can be regarded as typical representatives of all essentially different groups of finite linear-programming methods.

The reader who is interested only in the application of linear-programming may pass, after studying Chapter 1 (and §§ 1 and 2 of the Appendix), to the study of the various methods and computational schemes. On first reading one may omit §§ 4–6 and 8 of Chapter 4, § 9 of Chapter 5, §§ 2 and 4 of Chapter 6, and § 3 of Chapter 7.

To grasp the techniques of numerical solution of formalized linear-programming problems, it suffices to study §§ 2, 3, 5, 6, and 8 of Chapter 5, §§ 6 and 8 of Chapter 6, and §§ 5 and 6 of Chapter 7. The information presented in these sections together with that given in § 2 of Chapter 8 are sufficient for compiling a list of finite-method procedures for the solution of linear programming problems on digital computers.

Readers concerned with the development of theory and further perfection of the linear-programming methods should concentrate, besides Chapters 2 and 3, on those sections of Chapters 4 to 7 which deal with geometrical interpretation of the methods, the phenomena of degeneracy and cycling, and the specific features of problems with bilateral restraints.

The exercises given at the end of each chapter, besides numerical examples conducive to better understanding of the material, also include questions and problems which, for various reasons could not be included in the text.

The numeration of formulas in the book is autonomous for each section. The number of the formula consists of two numerals: the first indicates the number of the section, the second the serial number of the formula in the section. When reference is made to formulas from other chapters, the chapter number is also indicated.

The authors will be sincerely grateful to those readers who will take the trouble of sending in their remarks on the contents of this book.

October 1961 The Authors

Chapter 1

FUNDAMENTAL CONCEPTS OF LINEAR PROGRAMMING

§ 1. The subject of mathematical programming

1-1. The development of mathematics at the end of the 19th and the beginning of the 20th century was shaped mainly by the requirements of physical problems. Theoretical physics was responsible for the progress in differential geometry, vector and tensor analysis, the theory of differential and integral equations, probability theory and mathematical statistics as well as several other mathematical disciplines. The fundamental concept of the mathematical sciences in that period was continuity. The natural sciences and technology did not pose large-scale combinatorial problems to mathematicians, and furthermore, the available computational techniques were not able to provide effective methods for the solution of more complex problems.

In the last decade, however, various problems emerged in industrial organization, economic planning, automation of industry, and weapons control. A natural reaction to this trend was the development of cybernetics. Problems of power and technology control became as significant as problems of power and generation and technical development. It became obvious that the economic effect of effective control and planning methods, when applied on a large scale and on a suitably high level, may, in some cases, exceed the effect of a substantial increase in power output. New mathematical methods were required to analyze industrial processes, the operation of the nervous system of a living organism, the interactions between individuals and society and the quality of the military sciences. Although continuity, as before, is the central concept of contemporary mathematics, growing attention is given to discrete combinatorial problems.

The feasibility of solving numerically laborious mathematical problems within a reasonable time is now equally as important as proving difficult abstract propositions and deriving elegant formulas. Incidentally, the complicated modern problems of mathematics are not amenable, as a rule, to description in terms of formulas relating the given and the required quantities. In those cases where qualitative estimates are insufficient, it is better to try and develop algorithms which make it possible to calculate the result for each concrete system of initial data.

The role of physics as the purveyor of new problems and trends in mathematics has, thus, recently been replaced by cybernetics. The emergence and development of digital computational techniques led, of necessity, to the creation of new mathematical methods which, in their turn, extended the possibilities of mathematical machines.

1-2. Computers introduced into industry and control and employed in scientific research work open enormous vistas for developing various branches of science and for perfecting methods of industrial planning and automation. However, the applications of computational techniques cannot attain the required standard unless the problems are rigorously formulated and the processes described mathematically. The difficulty is often created not by problems of computer design, but rather by problems arising from the formal description of physical, economical, technical and other processes. Formulation of a problem is a necessary stage preceding its translation to the language of computers.

The general formulation of the problems of industrial control and planning can be described as follows.

The problems of control and planning often reduce to the choice of a particular system of parameters and functions. We shall refer to these parameters and functions as the control characteristics. To choose any particular set of control characteristics, we must clearly establish two points. First, we must decide what is a good system, and further what is the best system of planning and control. In other words, we should formulate and express in terms of the required characteristics a measure of quality, i. e., find a criterion specifying to what extent the plans being developed achieve the initial purpose of the process. Second, we must elucidate the operating conditions of the system and the ensuing restrictions to be satisfied by the computational characteristics.

For the present we are concerned with problems whose control characteristics are systems of parameters, rather than functions.

The problems of planning and control can thus be reduced to extremum problems of the following type.

It is required to determine the maximum of the function

$$f(x_1, x_2, \ldots, x_n) \tag{1.1}$$

subject to the conditions

$$g_i(x_1, x_2, \ldots, x_n) \leqslant 0, \qquad i = 1, 2, \ldots, m, \tag{1.2}$$

$$x_j \geqslant 0, \qquad j = 1, 2, \ldots, t \leqslant n. \tag{1.3}$$

The function $f(x_1, \ldots, x_n)$ is the quality measure of the solution. Conditions (1.2)–(1.3) are the restraints of the problem. Conditions (1.3), which are inherent for numerous problems in which the variables x_j — the control parameters — physically cannot take on negative values, should be separated from restraints of the type (1.2).

The restrictions imposed on the problem may also be expressed in terms of equalities. In the system (1.2) the equality $g_k(x_1, \ldots, x_n) = 0$ is represented by the two inequalities

$$g_k(x_1, \ldots, x_n) \leqslant 0,$$
$$-g_k(x_1, \ldots, x_n) \leqslant 0.$$

Extremum problems of the type (1.1)–(1.3) should be considered in the analysis of various problems of national-economy control and of industrial automation. Economic planning as well as military strategy and logistics also give rise to problems of this type.

The mathematical discipline concerned with problems of type (1.1)–(1.3) and with the development of methods of solution under various assumptions imposed on the functions f and g_i is called m a t h e m a t i c a l p r o g r a m m i n g.

The term mathematical programming is unsatisfactory since programming generally indicates the compilation of computational routines, i. e., programs for computers. Possibly a more suitable term describing the problems treated in this independent branch of applied mathematics would be mathematical planning. A still better description of the subject is methods of solution of conditional extremum problems. However, mathematical programming has been universally accepted in the literature and we shall therefore use it here.

The principal feature of mathematical-programming problems, which distinguish them from ordinary extremum problems of classical analysis, is the appearance of inequalities among the restrictions (1.2) and (1.3). The methods of solution of conditional-extremum problems with the aid of the so-called Lagrange multipliers do not apply here in their original form. The classical methods of differential calculus have been developed in order to determine the extremum points inside the domain of definition of the function. Lagrange's method could, therefore, be applied directly to the solution of problems (1.1)–(1.3) if it were known in advance which of the conditions (1.2), (1.3) reduce to equalities for the values of the arguments specified by the solution of the problem. However, generally speaking, no such information is available. If we confine the procedure to the techniques of classical analysis, we may indicate the following method for solving the mathematical-programming problem.

Let the functions f and g_i satisfy all the requirements necessary for solving conditional-extremum problems by Lagrange's method.

We first calculate the absolute extremum (maximum) of the function $f(x_1, \ldots, x_n)$, neglecting the conditions restricting the range of the variables, and then check whether the coordinates of the extremum point also satisfy conditions (1.2), (1.3). If these conditions are satisfied, the point is said to be the solution of the problem. If at least one of the conditions (1.2), (1.3) is not satisfied, Lagrange's method should be used to determine the conditional extremum of the function $f(x_1, \ldots, x_n)$ for one of the conditions. Then we check whether the arguments at which the conditional extremum is attained satisfy the remaining conditions of the problem. Analogous calculations are carried out under the assumption that the solution of the problem reduces to equalities any other condition among (1.2), (1.3), any pair of conditions, any triad of conditions, etc. In all these cases, when the extremum point falls within the domain specified by conditions (1.2), (1.3), we calculate the corresponding value of the quality measure of the solution, i. e., the function $f(x_1, \ldots, x_n)$.

It can be easily observed that this method of solving mathematical-programming problems requires the investigation of an enormous number of conditional extremum problems, even if there are relatively few restrictions. Generally, the solution of each problem of this type involves tedious calculations. Thus, using the existing computational techniques and computers it is impractical to solve problems of the type (1.1)–(1.3) by methods of classical analysis. Hence the necessity of developing special methods of solving these mathematical-programming problems.

1-3. Mathematical programming is subdivided into several disciplines, depending on the properties of the functions f and g_i. These have attained various levels of development. The simplest branches of mathematical programming have, by now, been perfected and methods of solution, including suitable computational techniques, have been developed. In more

complicated cases we have only rudimentary methods of solution. In some classes of mathematical-programming problems only qualitative results have been obtained.

The problems and methods of mathematical programming can be classified according to various features.

Depending on the type of the measure function and of the restrictions, mathematical programming is divided into linear and nonlinear programming. In linear-programming problems, f and g_i are linear functions of the variables x_j.

Among nonlinear conditional extremum problems there is a special class of problems classified as convex programming. In convex-programming problems, it is necessary to calculate the maximum of a concave function on a convex set*. Any local maximum of a concave function defined on a convex set is also its absolute maximum on that set. This proposition is the basis of all methods for solving convex-programming problems.

The solution of convex-programming problems is simplified if restrictions (1.2) can be represented as linear equalities or inequalities.

Among the problems of convex programming, relatively detailed methods are available for problems of quadratic programming which require the calculation of the vectors $X=(x_1, x_2, \ldots, x_n)$ satisfying linear equalities and inequalities and maximizing a sum of quadratic and linear forms:

$$Q(x_1, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_i x_j + \sum_{j=1}^{n} c_j x_j.$$

It is assumed that the quadratic form is nonpositive definite, i. e., Q is a concave function.

It is obvious that in all the preceding formulations the requirement of maximization of a function can be replaced by minimization. In this case, however, a concave measure function should be replaced by a convex measure.

Nonconvex problems of nonlinear programming have not yet been thoroughly investigated.

Mathematical programming is divided into branches which deal with optimal methods of planning and control under conditions of either total information or uncertainty, depending on whether the initial parameters of the problem are determined quantities or random factors. The analysis of some classes of mathematical-programming problems under uncertainty can be carried out by applying convex-programming techniques. This method is called stochastic programming.

In planning and control problems, the solution obtained at one stage often greatly limits the freedom of choice in subsequent stages. Accordingly, it is expedient to divide the problems and methods of mathematical programming into single-stage and multi-stage, or into static and dynamic problems.

In principle other classifications of the branches of mathematical programming are also possible. Within each of the branches it is possible to distinguish, in turn, between classes of problems requiring special methods of solution. Thus, for example, a large class of problems is associated with the application of certain standards. The required control parameters in these problems can assume only a limited number of discrete

* For the definition of convex and concave functions and of a convex set, see Appendix, § 3.

values. The study of these problems, which involve certain difficulties, gave rise to the so-called all-integer programming. All-integer programming is of special interest since numerous nonlinear nonconvex problems of mathematical programming can be reduced to linear-programming problems, subject to the requirement of integer solutions.

In some cases the initial parameters of the problem may vary within a certain interval. The so-called parametric programming is concerned with the variation of the parameters of the measure function and the restraints imposed on the solution.

Applied problems of mathematical programming with numerous variables and restraints often lead to memory overflow in modern digital computers. It became, thus, necessary to develop a new trend, the so-called block programming, which deals with the possibility of obtaining an exact or an approximate solution of a bulky problem from solutions of several particular problems with fewer variables and restraints.

The study of the features of the special classes of applied problems led to certain interesting qualitative and quantitative results. However, most of these results have not yet been unified.

1-4. Before concluding this section we find it worthwhile commenting on the formulation of mathematical-programming problems.

Mathematical analysis applies not to actual phenomena, but rather to certain mathematical models of these phenomena. These abstract models, naturally, do not encompass all, but only the most significant aspects of the phenomenon in question. The greatest responsibility lies in chosing those characteristics which are the most essential for the given problem and should, therefore, be included in the formulation of the mathematical model.

The phenomena being studied are not isolated. They are related to and interact with other phenomena of nature and life, which are possibly of no interest to the problem in question. When stating a problem, we must decide which of the relationships are to be neglected and which are to be replaced by certain restraints imposed on the required parameters. The thoroughness exercised in this stage will determine the advisability of using a complicated mathematical apparatus in the analysis of the problem and in the applicability of the solution.

In the formulation of planning and control problems special attention must be paid to the choice of the measure function of the solution for the corresponding mathematical-programming problem. The required planning and control method must, often, meet highly diverse, and often contradictory, requirements. As a rule, there does not exist a program or a control method maximizing (or minimizing) all the various criteria. The analysis of the results to be achieved by planning will bring out the most important index, which should then be optimized, and also yield the admissible limits of variation of the other criteria. The parameter being optimized is the quality measure. The admissible limits of variation of the other characteristics impose additional restraints on the problem.

An important stage in problem formulation is the choice of the variables which are to be taken as control parameters. The variables should be chosen so as to ensure the simplest possible form of the restraints and the measure function. Successful choice of the control parameters will determine the final amount of computational work necessary in the solution stage.

The problems of mathematical programming and, in particular, linear-programming problems to which the applied problems of planning and

control are reduced, involve, as a rule, numerous variables and restraints. The mathematical models, particularly models of phenomena which are studied for the first time, often do not reflect all the conditions. This limits the range of the problem variables. Some factors and limitations, which appear to be natural, are considered as self-evident and are not stated explicitly. If the solution of the problem is not attained by purely formal methods, reasoning in terms of concrete applications generally results in the following: the problem is subjected, at the solution stage, to additional restraints ensuing from its physical interpretation. However, an explicit solution of the problem is possible only if there are only a few variables and restraints. In practice, the mathematical-programming problems encountered in industrial and economic planning or in the control of technical and military operations are highly complicated, and can only seldom be completely analyzed. In these cases the problem must be carefully formulated and all the essential restraints, even if trivial, should be included in the mathematical model. For example, cases are known when solutions of least-cost diet problems, ensuring all the required nutrients, yielded quite unedible menus. This was due to the omission of edibility criteria from the mathematical model. There are sometimes cases when an analysis of mathematical-programming problems gives impracticable solutions: this may be so because in the formulation of the problem not all the limiting factors affecting the choice of the variables have been taken into consideration. The existence of additional restraints often emerges only after the causes of the impracticability of the solution have been analyzed.

The statement and the solution of applied problems often cannot be obtained in a single stage. Quantitative analysis of the solution generally points to the way for improvement of the model, with the purpose of attaining better agreement with the actual phenomenon. After the preliminary statement of the problem, it is expedient to obtain the formal solution for the simplest cases, when the optimal program is also known (exactly or approximately) from physical considerations. The analysis of deviations of the formal solutions from the expected result makes it possible to introduce adjustments to the formulation of the problem or, as may indeed occur, may modify our previous notions. Comparison of the formal solutions with the expected result must, however, be carried out keeping in mind that factors which are essential in a certain range of the initial parameters may become quite insignificant in another range.

Allowing for the additional factors emerging in the formulation stages, we may have to modify the choice of the control parameters to be determined. The most detailed consideration of the restraints imposed on the variables will not yield the solution of the problem if the mathematical model cannot be dealt with by existing methods. Serious problems of mathematical programming are, therefore, best formulated by specialists in the applied sciences working in cooperation with mathematicians.

§ 2. The object of linear programming

2-1. Linear programming is the best developed branch of mathematical programming. The scope and principles of solution of linear-programming

problems have been formulated fairly rigorously. We may say that at present linear programming constitutes an independent branch of applied mathematics which has reached a measure of perfection.

The object of linear programming is to calculate the extremum (maximum or minimum) of linear measure functions subject to the condition that the variables to be determined satisfy linear equalities or inequalities. Numerous planning and control problems of national economy, technology, strategy and logistics are stated in terms of linear programming. The necessity for immediate applicability of these problems is the reason for the considerable effort being made in developing suitable methods of solution.

The term linear programming is as unsatisfactory as the term mathematical programming. Nevertheless, we use this term here too since it is universally accepted.

We should again emphasize that the word "programming" is used here in the sense of the development of methods of solution of extremum problems, and the adjective "linear" indicates that the measure function and the restraints imposed on the unknown variables are linear functions.

The general linear-programming problem is stated as follows:

It is required to determine the maximum of a linear function of n variables $x_1, x_2, \ldots, x_n$

$$L = L(x_1, \ldots, x_n) = c_1x_1 + c_2x_2 + \ldots + c_nx_n \tag{2.1}$$

subject to the following restraints imposed on the variables $x_1, \ldots, x_n$:

$$\left.\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\ \cdots\cdots\cdots\cdots \\ a_{l1}x_1 + a_{l2}x_2 + \ldots + a_{ln}\,x_n = b_l, \end{array}\right\} \tag{2.2}$$

$$\left.\begin{array}{l} a_{l+1,1}x_1 + a_{l+1,2}x_2 + \ldots + a_{l+1,\,n}x_n \leqslant b_{l+1}, \\ \cdots\cdots\cdots\cdots\cdots \\ a_{r1}x_1 + \ a_{r2}x_2 + \ldots + a_{rn} \quad x_n \leqslant b_r, \end{array}\right\} \tag{2.3}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, t \quad (t \leqslant n). \tag{2.4}$$

The linear function (2.1), the quality measure of the variables in question, is generally called the linear form of the problem, and the set $(x_1, x_2, \ldots, x_n)$, satisfying conditions (2.2)–(2.4), is the domain of definition of the problem, or the domain of definition of its linear form.

Let us now introduce some additional definitions which will be rendered more precise in the following in the application to the various mathematical forms of the linear-programming problem*.

The matrix of the coefficients

$$A = \begin{Vmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ \cdot & \cdot & \cdot & \cdot \\ a_{r1} & a_{r2} & \ldots & a_{rn} \end{Vmatrix}$$

will be called the restraint matrix of the problem.

The columns of the restraint matrix, i. e., the vectors comprising the coefficients of the variables x_j, will be called the restraint vectors

$$A_j = (a_{1j}, a_{2j}, \ldots, a_{rj})^T.$$

(T, the transposition symbol, indicates that the vector A_j, whose components are arranged in a row, is a column vector.)

* The reader who is not familiar with the concepts of linear algebra used here will find the necessary explanations in the Appendix.

The vector, whose components are the right-hand sides of the problem conditions, will be called the constraint vector

$$B=(b_1, b_2, \ldots, b_r)^T.$$

Let

$$A_j'=(a_{1j}, a_{2j}, \ldots, a_{lj})^T, \quad A_j''=(a_{l+1,j}, a_{l+2,j}, \ldots, a_{rj})^T,$$

and, correspondingly,

$$B'=(b_1, b_2, \ldots, b_l)^T, \quad B''=(b_{l+1}, b_{l+2}, \ldots, b_r)^T.$$

Then

$$A_j=(A_j', A_j''),$$
$$B=(B', B''),$$

and the problem (2.1)–(2.4), can be rewritten in the following compact form:

Maximize the linear form

$$L(x_1, \ldots, x_n)=\sum_{j=1}^{n} c_j x_j \tag{2.5}$$

subject to the conditions

$$\sum_{j=1}^{n} A_j' x_j = B', \tag{2.6}$$

$$\sum_{j=1}^{n} A_j'' x_j \leq B'', \tag{2.7}$$

$$x_j \geq 0, \quad j=1, 2, \ldots, t \leq n. \tag{2.8}$$

A vector $X=(x_1, x_2, \ldots, x_n)$, satisfying conditions (2.2)–(2.4), or equivalently (2.6)–(2.8), will be called a feasible program of the problem.

The program $X^*=(x_1^*, \ldots, x_n^*)$, maximizing the linear form (2.1) is called the optimal program, or the solution of the problem.

The terms program and optimal program originating in economic applications are also retained in the general linear-programming problem.

In what follows we shall operate with the matrix form of linear-programming problems.

Let us introduce the matrices

$$A'=(A_1', A_2', \ldots, A_n'),$$
$$A''=(A_1'', A_2'', \ldots, A_n')$$

and the vectors

$$C=(c_1, c_2, \ldots, c_n),$$
$$X'=(x_1, x_2, \ldots, x_t),$$
$$X''=(x_{t+1}, \ldots, x_n),$$
$$X=(X', X'')^T.$$

The general linear-programming problem in these new notations is formulated as follows:

Calculate the vector $X=(X', X'')^T$ maximizing the linear form

$$L(X)=CX \tag{2.9}$$

subject to the conditions

$$A'X=B', \tag{2.10}$$
$$A''X\leq B'', \tag{2.11}$$
$$X' \geq 0. \tag{2.12}$$

The methods and algorithms of linear programming are discussed in what follows mainly for problems requiring maximization of the linear form. This assumption obviously does not detract from the generality of the discussion. The problem involving minimization of the linear form reduces to the maximization problem when the signs of all the coefficients c_j are reversed.

In the analysis of the conditions (2.2)–(2.4), three cases may arise:

(a) conditions (2.2)–(2.4) are inconsistent, i. e., there does not exist a set of numbers $x_1, x_2, \ldots, x_n$, satisfying all the conditions of the problem;

(b) conditions (2.2)–(2.4) are consistent, but the domain defined by these conditions is not bounded, i. e., there exist sets of numbers $x_1, x_2, \ldots, x_n$ satisfying conditions (2.2)–(2.4) and containing individual variables of arbitrarily large values;

(c) conditions (2.2)–(2.4) are consistent and the domain defined by these conditions is bounded.

Let us now introduce the concept of solvability of a problem which will be useful in the subsequent discussion.

A linear-programming problem is said to be *solvable* if there exists a set of numbers $(x_1, x_2, \ldots, x_n)$ $(x_j < \infty)$, which satisfy all the restraints (2.2)–(2.4) and, depending on the problem requirements, maximize or minimize the linear form (2.1).

Unsolvability of a problem may be due to either inconsistency of the problem conditions (case (a)), or to the lack of boundedness of the linear form in its domain of definition (case (b)). Observe that case (b) does not necessarily result in an unsolvable problem. A linear form may be bounded even in an unbounded domain.

2-2. In the following (Chapter 2) we shall show that the domain of definition of the linear form of a linear-programming problem is a convex polyhedral set in n-dimensional space of the variables x_j, and the extremum of the linear form is attained at the vertices, which are finite in number. The vertex coordinates of the polyhedral set of problem conditions satisfy the equalities corresponding to linearly independent restraints of the type (2.3)–(2.4). Therefore, the theoretical method of solution of the mathematical-programming problem outlined in 1-2 is considerably simplified in the linear case. There is no need to look for extrema for $1, 2, \ldots, n-1$ constraining equalities. The coordinates of the extremum point in a linear-programming problem always satisfy n independent constraining equalities. However, even in linear-programming problems the theoretically feasible method is impracticable. Even for comparatively small n, r and t (cases corresponding to relatively simple applied problems), there are several billion vertices of the corresponding polyhedral set. This means that an unordered processing of the vertices, with the purpose of finding the point at which the linear form is maximized or minimized, is a problem which cannot be tackled even on the fastest modern computer.

To get an idea of the laboriousness of the computations entailed in random processing of the vertices, consider one of the classical problems of linear programming — the choice problem — for which we can easily calculate the number of vertices of the corresponding polyhedral set. In the choice problem we are given a tableau of n rows and n columns. It is required to choose one element from each row and each column so that their sum is a minimum. This problem, which has diverse practical applications,

is a linear-programming problem. The number of vertices of the corresponding polyhedral set (in the choice problem this set is bounded and is consequently a polyhedron) equals $n!$

The direct solution of the choice problem thus involves the comparison of of $n!$ quantities. To calculate the value of the linear form at each of the polyhedron vertices, we should carry out n summations. For $n>15$ all the operations required to solve the problem cannot be carried out within a reasonable time either on existing or on projected computers.

Even using Stirling's formula

$$n!\approx\left(\frac{n}{e}\right)^n\sqrt{2\pi n},$$

for $n=20$, the number of polyhedron vertices, $n!$, is greater than $2\cdot 10^{18}$. A computer performing 10 million operations per second (there are still no such computers available) would require 5000 years to process the polyhedron vertices of this relatively simple problem.

For $n=30$, the number of vertices of the domain of definition of the choice problem is greater than 10^{31}. It seems that it would be easier to count the grains of sand on the earth than to process the polyhedron vertices of this problem.

In practice solutions are required for problems similar to the choice problem with n considerably larger than 20 and 30. The preceding examples clearly show that special methods must be developed for the solution of linear-programming problems.

As we shall see in the following, the various methods available for solving linear-programming problems amount to ordering the processing of the vertices of the polyhedral sets defined by the constraints of the given problem, or some other problem associated with the original one. Each method of linear programming has a characteristic vertex-optimality criterion. The optimality criterion makes it possible to establish whether the extremum of the linear form is attained at the particular vertex, without comparing the values of L at all the vertices of the polyhedral set. The optimality criteria result from the property that in linear-programming problems every local extremum is also an absolute extremum.

If the vertex considered does not correspond to the extremum of the linear form, the next vertex should be taken. Methods of linear programming direct this processing in such a way that each successive vertex is closer to the extremum than the preceding one. Methods of linear programming also make it possible to establish unsolvability of a problem, if it is in fact unsolvable.

Interestingly enough, ordered processing of vertices makes it possible to solve the choice problem after about n^3 elementary operations. Existing computers then require less than a minute for solving the choice problem for $n=20$ and $n=30$.

§ 3. Linear-programming problems

In this section we briefly consider some applied problems falling within the scope of the model (2.1)–(2.4).

In 3-1, where the allocation problem is considered, we outline

the characteristic cases in which the solution of the linear-programming problem does not require special techniques and cumbersome computations.

In 3-2 we discuss an important particular linear-programming problem, the so-called transportation problem. The transportation problem is a natural extension of the allocation problem for a single origin and a homogeneous commodity.

In 3-3 we formulate the optimal weapons problem. In this case, translation of the measure function and the restraints into formal language immediately yields a problem of the type (2.1)--(2.4). Unlike this problem, the multi-component equipment problem, considered in 3-4, reduces to a linear-programming problem only after certain (although simple) artificial manipulations.

3-1. We consider the problem of optimal allocation of a homogeneous commodity to an industrial center, the commodity being, e. g., potatoes or coal. The commodity in question can be supplied from n origins.

Let x_i be the amount delivered to the destination from the i-th origin, and c_i the overall cost of producing and shipping a unit amount from the i-th origin. Then the cost of the product shipped from the i-th origin to the destination is $c_i x_i$, and the cost of the product delivered at the destination from all the production origins is

$$L = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n. \tag{3.1}$$

We must allocate the supply, i. e., choose the x_i in such a way as to ensure minimum delivery cost of the product. The following conditions must be taken into consideration in the process: the actual requirement, specified by b, should be satisfied exactly. Hence, the required variables should satisfy the condition

$$x_1 + x_2 + \ldots + x_n = b. \tag{3.2}$$

The production of the commodity at the i-th origin is limited by b_i, and the carrying capacity of the shipping routes, which can be chartered for shipping the product from the i-th origin, is limited by d_i.

Let β_i be the smaller of the two numbers b_i and d_i,

$$\beta_i = \min(b_i, d_i), \quad i = 1, 2, \ldots, n.$$

The required variables x_i should thus satisfy the restraints

$$x_i \leqslant \beta_i, \qquad i = 1, 2, \ldots, n. \tag{3.3}$$

Since a negative shipment has no valid interpretation for the problem as stated, we restrict the x_i:

$$x_i \geqslant 0, \quad i = 1, 2, \ldots, n. \tag{3.4}$$

If the commodity from the i-th origin can be sent by two different types of transport, the i-th origin should be considered as two origins with different production costs.

We have thus obtained a linear-programming problem. It is required to minimize the linear form L (3.1), subject to the linear restraints (3.2)–(3.4).

This problem has a characteristic property which makes it possible to compute easily the values of x_i for which the cost of the commodity at the destination is minimized.

It is obviously advisable to receive as much of the commodity as possible

from those origins for which the c_i (the cost of production and shipment per unit commodity) are small. Let us renumber the production origins in order of increasing c_i. We now have

$$c_1 \leqslant c_2 \leqslant \ldots \leqslant c_n.$$

By assumption, the destination may receive from the first origin (the one supplying the cheapest product) not more than β_1 units of the commodity. If the total requirement of the destination, measured by b, does not exceed the resources of the first origin, i. e., if $b \leqslant \beta_1$, it would be best to satisfy the entire requirement from the surpluses of the first origin. In this case

$$x_1 = b, \quad x_2 = x_3 = \ldots = x_n = 0.$$

If, however, $b > \beta_1$, it would be expedient to "ship most at the least cost", i. e., to take $x_1 = \beta_1$, satisfying the remaining part of the requirements $(b - \beta_1)$ from the remaining origins.

We have here obtained an allocation problem analogous to the preceding one, the difference being that the requirements of the destination are specified by $b - \beta_1$ and the number of production origins is now $n-1$. Here again two cases are possible. In the first case $(b - \beta_1) \leqslant \beta_2$. Then the optimal allocation system is specified by the equalities

$$x_2 = b - \beta_1, \quad x_3 = x_4 = \ldots = x_n = 0$$

($x_1 = \beta_1$ has been previously determined).

In the second case, where $(b - \beta_1) > \beta_2$, it is expedient to take $x_2 = \beta_2$ and to proceed with the problem entailing fewer origins $(n-2)$ and smaller requirements $(b - \beta_1 - \beta_2)$.

The preceding reasoning leads to the following procedure for solving the problem.

From b we successively subtract the numbers $\beta_1, \beta_2, \ldots$ Two cases are possible:

a) $b - \beta_1 - \beta_2 - \ldots - \beta_n > 0$;
b) $b - \beta_1 - \beta_2 - \ldots - \beta_n \leqslant 0$.

In the first case the requirements of the destination cannot be completely satisfied. The demand exceeds the overall supply potential of all the origins.

In the second case, the demand can be completely satisfied.

Let us determine the index k from the condition

$$b - \beta_1 - \ldots - \beta_k \geqslant 0,$$
$$b - \beta_1 - \ldots - \beta_k - \beta_{k+1} < 0.$$

The optimum allocation system is then determined by:

$$x_1 = \beta_1, \quad x_2 = \beta_2, \quad \ldots, \quad x_k = \beta_k;$$
$$x_{k+1} = b - \beta_1 - \beta_2 - \ldots - \beta_k; \quad x_{k+2} = \ldots = x_n = 0.$$

Thus, in the optimal allocation program the first k origins are relieved of all their surpluses, and the subsequent $n-(k+1)$ origins do not enter the program at all.

Observe that we obtained the solution of this problem by elementary reasoning. Unfortunately, this is not the rule, but rather the exception in linear-programming problems. The simplicity of the solution arises from the fact that the problem conditions contain only a single restraint imposed on all the variables. All the other conditions only restrict the range of each of the variables separately.

Let us now make the allocation problem for a single destination and a homogeneous commodity more complicated by introducing an additional restraint.

For example, let the loading time of the conveyance intended for shipping the commodity be limited and, moreover, let the mechanization of the loading operations be different in different origins. Let a_i be the time required at the i-th origin for loading one transport unit and let T be the limit imposed on the overall loading time in the network. It is natural to assume that the number of transport units y_i, required for shipping the commodity from the i-th origin is proportional to the bulk of the commodity shipped. The idling time of the transport at the i-th origin is given by $a_i y_i = \alpha a_i x_i$, and the total idling time of the network by $\alpha(a_1x_1 + a_2x_2 + \ldots + a_nx_n)$.

The allocation problem is now formulated as follows. Minimize the linear form (3.1) subject to (3.2)–(3.4) and satisfying the additional restraint

$$\alpha(a_1x_1 + a_2x_2 + \ldots + a_nx_n) \leqslant T.$$

Similar considerations as in the above will not provide a solution here. In this case and, moreover, in those cases where new restraints must be taken into consideration, allocation planning requires special methods which are the subject matter of linear programming.

Consider, for instance, the allocation problem for an inhomogeneous commodity, i. e., vegetables or fuel. In this case we can indicate the conditions of interchangeability, the minimum and the limiting requirements of each individual product, e. g., separately for potatoes and cabbage, or coal and oil. Interchangeability should be characterized by a coefficient indicating how many units of one commodity are equivalent to one unit of the other. The concept of equivalence can be defined differently in different problems. In our problem, we may assess the interchangeability of the commodities from, for example, the calory content of the vegetables or the fuel.

Retaining the notations adopted in the previous case (homogeneous commodity) we now formulate the problem mathematically.

Minimize the linear form

$$L = c_1^{(1)}x_1^{(1)} + \ldots + c_n^{(1)}x_n^{(1)} + c_1^{(2)}x_1^{(2)} + \ldots + c_n^{(2)}x_n^{(2)}$$

subject to the conditions

$$b_{\min}^{(1)} \leqslant x_1^{(1)} + \ldots + x_n^{(1)} \leqslant b_{\max}^{(1)},$$
$$b_{\min}^{(2)} \leqslant x_1^{(2)} + \ldots + x_n^{(2)} \leqslant b_{\max}^{(2)},$$
$$0 \leqslant x_i^{(1)} \leqslant \beta_i^{(1)},$$
$$0 \leqslant x_i^{(2)} \leqslant \beta_i^{(2)},$$
$$\mu_1(x_1^{(1)} + \ldots + x_n^{(1)}) + \mu_2(x_1^{(2)} + \ldots + x_n^{(2)}) = \mu.$$

The superscript enumerates the commodity. If some of the n origins produce only one of the commodities being considered, the corresponding values of $x_i^{(s)}$ are taken as zero from the outset. The last equality among the restraints is the interchangeability condition. It may be interpreted, for example, as follows: $\mu_1(x_1^{(1)} + \ldots + x_n^{(1)})$ is the number of calories released when $x_1 + \ldots + x_n$ tons of coal are burnt, $\mu_2(x_1^{(2)} + \ldots + x_n^{(2)})$ is the quantity of heat released by oil, and μ is the required quantity of heat. Finally, there is no difficulty in generalizing the statement of the problem for the case of

allocating an arbitrary amount of inhomogeneous commodities to several destinations.

3-2. The allocation of a homogeneous product from m origins to n destinations, which is the subject of the so-called transportation problem, constitutes one of the first applications of linear programming. Numerous economic and military problems can be reduced to the formal outline of the transportation problem. The essence of the transportation problem is as follows.

There are m origins of a homogeneous product $A_1, A_2, \ldots, A_m$. The production capacity of the origin A_i is a_i units of the commodity. The entire supply of the commodity is consumed in n consumption centers $B_1, B_2, \ldots, B_n$. The consumption capacity of the destination B_j is b_j units of the commodity.

It is necessary to plan the supply to the destinations B_j $(j=1, 2, \ldots, n)$, from the origins A_i $(i=1, 2, \ldots, m)$, so that the total transportation costs are minimum. It is assumed that shipment can be arranged from any origin to any destination, the shipment charges being proportional to the amount shipped. The shipment schedule should fill the requirements of all the consumption centers without leading to overstocking of the production sources. Production and consumption are obviously assumed to balance, i. e., the total production capacity equals the total consumption capacity.

The formal description of the problem requires the following notations. Let x_{ij} be the number of commodity units to be shipped from the i-th origin to the j-th destination, and c_{ij} the costs incurred in the shipment of one unit of the commodity from A_i to B_j.

In the transportation problem it is required to find such values of x_{ij} which minimize the total shipment costs

$$L=\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

subject to the conditions

$$\sum_{j=1}^{n} x_{ij}=a_i, \qquad i=1, 2, \ldots, m, \tag{3.5}$$

$$\sum_{i=1}^{m} x_{ij}=b_j, \qquad j=1, 2, \ldots, n, \tag{3.6}$$

$$x_{ij}\geqslant 0. \tag{3.7}$$

Conditions (3.5) indicate that from the origin A_i $(i=1, 2, \ldots, m)$, a_i units of the commodity, i. e., the entire production capacity of A_i, are shipped to all the consumption centers.

Conditions (3.6) indicate that the consumption center B_j receives from all the production sources b_j units of the commodity, i. e., an amount equal to the consumption capacity of B_j.

Inequalities (3.7) indicate that shipments are arranged only from the production sources to the consumption centers; there are no return shipments.

This statement implies the equality of production and consumption. Summing the equalities (3.5) over i and (3.6) over j, we obtain

$$\sum_{i=1}^{m} a_i=\sum_{j=1}^{n} b_j.$$

The transportation problem is one of the most important problems of linear programming. The specific conditions of the transportation problem

made it possible to simplify essentially the general methods of the solution of linear extremum problems as applied in this case.

3-3. The problem of optimal weapon systems given here is a slight modification of the discussion presented in /80/.

Weapons of various classes can be applied to achieve the same purpose, e. g., to destroy the same objects. It is required to determine in what amounts should weapons of different classes be produced to ensure the maximum effectiveness of the weapons system for given resources and limited funds. From this statement of the problem we consider, for instance, the allocation of funds between production orders placed for strategic aircraft of various types, missiles, and nonguided strategic rockets. The same problem can be easily formulated in terms of the optimal antitank or air-defence weapons system.

One of the principal assumptions in this problem is that each weapon is restricted to a definite group of objectives.

Let x_j be the number of weapons of type j, and p_j be the average number of targets hit by a single weapon of type j. From our assumption, the quality measure of the weapons system — the mathematical expectation of the number of eliminated objectives — can be written in the form

$$M = p_1x_1 + p_2x_2 + \ldots + p_nx_n. \tag{3.8}$$

Let the cost of a single weapon of type j be r_j. Then, the restriction imposed on the number of components in the weapons system by the assigned funds, b, can be represented by the inequality

$$r_1x_1 + r_2x_2 + \ldots + r_nx_n \leqslant b. \tag{3.9}$$

A weapon is a multi-component piece of equipment whose production is limited not only by capital investments, but also by the necessity of employing special apparatus and materials supplies of which are limited.

The demand of various factors (raw materials, machine shop, man-hours, etc.) for weapons of different types is different. Let a_{ij} be the number of units of the factor i required for the production of one unit weapon of type j, and b_i the restraint imposed on this factor. Then, the additional conditions to be imposed on the choice of x_j are written in the form

$$\left.\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leqslant b_1, \\ \cdots\cdots\cdots\cdots\cdots \\ a_{s1}x_1 + a_{s2}x_2 + \ldots + a_{sn}x_n \leqslant b_s. \end{array}\right\} \tag{3.10}$$

Here s is the number of limiting factors.

Beside the preceding conditions, weapons production is limited by storage and maintenance factors, availability and training of qualified service personnel, etc. Some of the required variables (let these be the first k of the system), are therefore additionally restrained by conditions of the form

$$x_j \leqslant \alpha_j, \quad j = 1, 2, \ldots, k \leqslant n. \tag{3.11}$$

Finally, all the variables (the number of weapons of each given type) should obviously be nonnegative, i. e.,

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n. \tag{3.12}$$

Generally speaking, the physical interpretation of the variables imposes another condition, namely that the x_j should be integers. Yet departure from this requirement in problems involving fairly large x_j does not lead,

as a rule, to essential deviations from the optimum. With fractional x_j, the solution should be considered as the closest integer to x_j.

We observe that the choice of the optimal weapons system requires maximization of the linear form (3.8), subject to the linear conditions (3.9)–(3.12). This is obviously a linear-programming problem.

If the choice of the optimal weapons system were restrained only by conditions (3.9), (3.11) and (3.12), the solution of the problem could be derived from the simple considerations shown in 3-2. In the general case, however, more complicated methods are needed.

3-4. Consider the multi-component equipment problem, concerned with planning the production of pieces of equipment comprising n elements. For example, this may be a system consisting of n instruments, or an instrument made up of n components, etc.

Orders may be placed with m distinct enterprises possessing different machine-tool resources or different potentials. It is necessary to determine the distribution of orders between the enterprises which would ensure the production of the maximum number of composite systems within a scheduled period. Analogous problems are considered when the production of a single instrument is planned, with the various components manufactured on different machines of the enterprise or shop. Optimal occupancy of the equipment should ensure the maximum release of instruments.

Let unit time be taken as the period for which the entire working program is scheduled. Let the i-th enterprise (the i-th machine) be capable of producing a_{ij} elements of equipment of type j. Assume that all $a_{ij} > 0$, i. e., assume that each component can be manufactured on any of the machines. Let x_{ij} be the time during which the i-th contractor (enterprise or machine) is occupied with manufacturing elements of type j. The number of components of type j released by all the contractors in unit time is

$$a_{1j}x_{1j}+a_{2j}x_{2j}+\ldots+a_{mj}x_{mj}, \qquad j=1,\ 2,\ \ldots,\ n. \tag{3.13}$$

Only assembled products can be delivered. The number of pieces of equipment which can be assembled from the manufactured components is limited by the class of elements of which fewest are available. In other words, the number of assembled units L, produced in unit time equals the least of the elements of (3.13) calculated for various j:

$$L=\min\,[a_{1j}x_{1j}+a_{2j}x_{2j}+\ldots+a_{mj}x_{mj}].$$

The optimality criterion for order placement and equipment occupancy is that the time intervals x_{ij} be so chosen that each enterprise (each machine) has no idle time and that L be maximized. In formal language this means that L must be maximized subject to the conditions

$$\left.\begin{array}{c} x_{i1}+x_{i2}+\ldots+x_{in}=1, \\ x_{ij}\geqslant 0, \quad i=1,\ 2,\ \ldots,\ m, \quad j=1,\ 2,\ \ldots,\ n. \end{array}\right\} \tag{3.14}$$

The first group of conditions requires that the sum of the times taken by each contractor in manufacturing the assembly elements be equal to unity, i. e., the time to which the entire program has been scheduled. This indicates that the enterprises (machines) should have no idling time.

The second group of conditions indicates that the occupancy time of each contractor with each element cannot be negative.

The problem of maximizing the minimum at which we have arrived is easily reduced to a simple linear-programming problem. Indeed, we

readily observe that the maximum number of assembled units can be released only if the occupancy of the contractors ensures that the various components are all produced in equal numbers, i. e., the solution of the problem must satisfy the conditions

$$\sum_{i=1}^{m} a_{i1}x_{i1}=\sum_{i=1}^{m} a_{i2}x_{i2}= \ldots =\sum_{i=1}^{m} a_{in}x_{in}.$$

Indeed, had one of these sums been greater than all the others, with $a_{ij}>0$ we could have increased the minimum sum and thus increased the number of assembled units by slightly decreasing the excessive components, and all this without violating (3.14). This, however, would indicate that it is not the optimal system of order placements which has been chosen.

Hence, the preceding problem may also be formulated as follows:

Maximize the linear form

$$L=a_{11}x_{11}+a_{21}x_{21}+\ldots+a_{m1}x_{m1}$$

subject to the conditions

$$\begin{gathered}
a_{11}x_{11}+a_{21}x_{21}+\ldots+a_{m1}x_{m1}=a_{12}x_{12}+a_{22}x_{22}+\ldots+a_{m2}x_{m2}= \\
=a_{13}x_{13}+a_{23}x_{23}+\ldots+a_{m3}x_{m3}= \\
\ldots\ldots\ldots\ldots\ldots\ldots \\
=a_{1n}x_{1n}+a_{2n}x_{2n}+\ldots+a_{mn}x_{mn}, \\
x_{11}+x_{12}+\ldots+x_{1n}=1, \\
x_{21}+x_{22}+\ldots+x_{2n}=1, \\
\ldots\ldots\ldots\ldots \\
x_{m1}+x_{m2}+\ldots+x_{mn}=1, \\
x_{ij}\geqslant 0, \quad i=1,\ 2,\ \ldots,\ m, \quad j=1,\ 2,\ \ldots,\ n.
\end{gathered}$$

We have thus obtained a linear-programming problem.

§ 4. Brief historical survey

Mathematical programming developed out of practical needs some twenty years ago.

The most developed and comprehensive branch of mathematical programming is linear programming. Works dealing with particular problems of linear programming date back to the beginning of the thirties. In 1931, for example, in Hungary, Egervary /123/ published a paper on the assignment problem, a particular case of the transportation problem. It is noteworthy that later the results of this paper were used by several authors to develop a highly effective method for the solution of the transportation method (see /70, 71, 111, 112/). In the literature this method is known as the *Hungarian method*.

The term *linear programming* was coined in America in the mid-forties (the first American paper on a particular problem of linear programming was published in 1941 /116/). In the Soviet Union, studies in this field began a little earlier. By the end of the thirties several essential results in linear programming were established by L. V. Kantorovich /61, 62, 66/*. In particular, in /61/ the powerful method of decision multipliers, used in particular when dealing with dual problems, was advanced and applied to some particular problems.

Decision multipliers are also very useful in formulating optimality tests (criteria) of linear problems. The set of algorithms entailing decision multipliers is sometimes referred to as the *method of decision multipliers*. Paper /61/ deals with one of the forms of the method of decision multipliers involving motion over the boundary of the cone of the given problem (the *method of successive residue contraction*). A geometrical interpretation of linear-programming problems is also given. A geometrical description of the method of residue contraction was given by G. Sh. Rubinshtein /94, 95/.

In 1956 Dantzig, Ford, and Fulkerson /52/, proceeding from the aforementioned Hungarian method, developed a general linear-programming method which differs from the method proposed by Kantorovich

* Paper /66/, published in 1949, was completed by L. V. Kantorovich and M. K. Gavurin in 1940. Publication was delayed because of the war. Paper /63/ published in 1942 contains a reference to this work.

in /61/ in insignificant details only. In a joint work Kantorovich and Gavurin /66/ proposed, in application to the transportation problem, another form of the method of decision multipliers involving motion outside the problem cone (the potential method).

The first American work on general problems of linear programming /40/ was published by Dantzig in 1949. It presents the main concepts of the simplex method (the first algorithm) in application to nondegenerate linear-programming problems. A method similar to the simplex was proposed by S. I. Zukhovitskii in application to problems dealing with the best Chebyshev approximation (see /58, 59, 60/).

Charnes in /117/ gives a procedure for application of the simplex method in degenerate cases (see also Dantzig, Orden, and Wolfe /50/).

The simplex method, its computational procedures, and various revisions were further considered by Dantzig and other authors (/41, 50, 51, 78, 85, 14/).

In 1954 Lemke /73/ proposed another general linear-programming method — the dual simplex method. In Lemke's method the solution of the dual problem by the simplex method is presented in terms of the primal problems.

Similar to Lemke's method is the method of leading variables proposed by Beale in /5/. Other available methods of linear programming include the gradient method /124/ and Frisch's method of logarithmic potential /113/, both closely related, the double description method /83/, Motzkin's relaxation method /82/, Tompkin's projection method /105/, and others. These methods are less effective than the three above mentioned general linear-programming methods, and their modifications.

Another group of methods comprises the so-called iterative methods adapted from numerical methods of solution of rectangular games. The equivalence of the programming problem and the game problem proved in /43/ makes it possible to solve linear-programming problems by Brown's method /9, 91/, Neumann's method /84/, and the method of differential equations proposed by Brown and Neumann in /10/. The iterative method developed by Bulavskii /11/ is not quite to the point.

From experience acquired till now in the solution of linear-programming problems, it seems that the iterative method is inferior to other methods.

Qualitative results and the discussion of the relationship between linear programming and the theory of games can be found in /25, 38, 43, 44, 49, 104, 107/.

Parallel with general methods of linear programming, methods have been developed for the solution of certain particular problems, such as the transportation problem, the assignment problem, the traveling-salesman problem, the timetable problem, the interindustry problem, etc. The most important results on methods of solution of the transportation problem and its various generalizations were published in /1, 22, 31, 42, 66, 76, 132/.

Of some interest in practical applications is the generalization of general and particular linear-programming methods to problems whose variables are restrained on two sides and whose functions are convex and piece-wise linear /26, 119, 130/. The computational procedures developed for these problems are very close to the algorithms of solution of the corresponding linear problems restrained on one side.

In recent years several works have been published on linear-programming problems where physical interpretation of the variables shows that they should be integers. There is a tendency to collect such problems into an independent branch of mathematical programming, the so-called integer programming. The main results in this field were published in /32, 33, 34, 35, 46, 79/.

Practical linear-programming problems contain a great many variables and restraints. Only modern computers can cope with the laborious computational work involved in the solution of these problems. Methods and techniques of computations on digital computers are discussed in /56/ and /37/. The experience acquired in solving linear-programming problems on analog installations is described in /122/.

Owing to the limited operative memory of modern computers there arose a need for the so-called block programming. The reader is referred to /45/ and /48/ as the main sources of this field.

In the formulation of planning and control problems we often lack the detailed information necessary for unique determination of the functional parameters and the restraints. There are three possible ways proposed for overcoming the difficulties involved. First, studying the effect of variations of the linear-form coefficients and of the restraints on program optimality /81/. Second, works on the so-called parametric programming /18, 19/ propose methods which, without substantially increasing the volume of computations, make it possible to give solutions for an entire range of variation of one or several parameters on which the linear-form coefficients and the restraints depend in a known manner. The third and last way defines a new branch of mathematical programming — the so-called stochastic programming or programming under uncertainty which deals with the development of effective methods of planning for given statistical characteristics of the problem conditions /47, 75, 87/.

Linear programming is a branch of applied mathematics. However, works on continuous analogs of

linear programming /53, 68, 85/ make it possible to employ the concepts of linear programming in various branches of mathematics (theory of best approximations, theory of moments, etc.). The application of duality to the theory of best approximation is discussed in /27, 28, 29, 30, 114/.

Soviet researchers have acquired considerable experience in application of linear programming to the solution of practical problems /55, 67, 99, 100, 101, 106/. Kantorovich's book /65/ and collections /97/ and /98/ edited by V.S. Nemchinov greatly promoted the application of linear programming in economic research. A large number of different applications of linear programming to industrial and economic planning problems was presented at the conference on application of mathematics to economics, held in Moscow in April 1960 /106/. Outside the USSR linear programming has found wide military applications /77, 80, 131/ and economic applications, in particular in questions of shop, plant, intracompany, and trade planning. A survey of applications of linear programming in the USA is given by Gass /18/. This book also gives a detailed list of references on applications of linear programming to economic and military problems.

Much less developed are the methods of analysis of nonlinear problems of mathematical programming.

Kuhn and Tucker /72/ were the first to deal with nonlinear programming in their extension of Lagrange multipliers to nonlinear extremum problems with inequality restraints. Qualitative aspects of nonlinear programming are also discussed in /102, 103, 108, 127/.

Numerical methods of nonlinear programming were mainly developed to include local extrema. These methods are clearly of much interest mostly in problems where it is known a priori that the local and the absolute extrema coincide.

Convex programming problems (i.e., problems with a convex functional whose restraints define a convex set in the space of variables) constitute the largest class of conditional-maximum problems with this property. These problems can be approximated with any degree of accuracy by linear problems. General methods of convex programming are discussed in /7, 57, 93, 109, 125, 126/. Problems of quadratic programming maximizing a sum of nonpositive definite quadratic and linear forms subject to linear equality and inequality restraints have been developed to a greater extent than all other problems of convex programming /8, 16, 115/. There are also several works dealing with methods of solution of special problems of convex programming. Among these problems we have, for example, the transportation post with nonlinear dependence of shipping cost on shipment volume /4, 20/. Convex-programming algorithms are essentially simplified if the problem functional is a sum of separable convex functionals /26, 119/.

In the solution of mathematical-programming problems with many local extrema considerable difficulties arise. Until now there are no sufficiently effective methods of solution of these methods. Certain progress in this field is due to the so-called trench method proposed by I.M. Gel'fand and M.L. Tsetlin /20/. There are also suggestions to apply Monte-Carlo methods to the solution of mathematical-programming problems with many local extrema.

Nonlinear problems with relatively few variables and restraints can be investigated by means of electrical analogs /54/.

Mathematical programming is a rapidly developing new science. Results, methods, and algorithms of mathematical programming often appear in various periodicals. New directions in this science are often mentioned not in mathematical works, but in publications on applied subjects. This historical survey, therefore, is not comprehensive. The reader wishing to become better acquainted with non-Soviet sources on mathematical programming is referred to Rohde's bibliography /92/ and also to annotated bibliographies in /90/ and /12/. A fairly comprehensive bibliography will be found in /96/.

§ 5. Canonical form of linear-programming problems

5-1. All the examples considered in § 3 fall under the general heading of linear-programming problems. However, the linear form and especially the restraints in the various problems differ essentially. Numerous examples can be given to show this diversity in the form of linear-programming problems. In some cases the required variables depend on one subscript, in others, on two. In some problems the restraints are specified by equalities, in others, by inequalities. Certain applied problems even lead to mixed restraints, some of which are linear equations, and some, linear inequalities. Not in all problems is nonnegativity of all the variables specified. This diversity of restraints requires that special methods be

developed for solving distinct classes of problems. Thus, the study of the general properties of linear programming and the development of general methods and computational algorithms becomes more difficult. It is, therefore, natural to consider the possibility of reducing any linear-programming problem to some general form.

We shall say that a linear-programming problem is given in the canonical form if it is stated as follows.

Maximize (minimize) the linear form

$$L=\bar{c}_1x_1+\ldots+\bar{c}_{\bar{n}}x_{\bar{n}}$$

subject to the conditions

$$\begin{gathered}\bar{a}_{11}x_1+\ldots+\bar{a}_{1\bar{n}}x_{\bar{n}}=\bar{b}_1,\\ \ldots\ldots\ldots\ldots\\ \bar{a}_{m1}x_1+\ldots+\bar{a}_{m\bar{n}}x_{\bar{n}}=\bar{b}_m,\\ x_1\geqslant 0,\quad\ldots,\quad x_{\bar{n}}\geqslant 0.\end{gathered}$$

The linear-programming problem, written in the general form (2.1)–(2.4), can be reduced to canonical form.

We introduce in (2.1)–(2.4) the additional nonnegative variables

$$\begin{gathered}x_{n+1},\ x_{n+2},\ \ldots,\ x_{n+(r-l)},\\ x_i\geqslant 0\ \text{ for }\ i=n+1,\ \ldots,\ n+(r-l).\end{gathered}$$

The restraints (2.2), (2.3) are then equivalent to the following:

$$\begin{array}{lr}a_{11}x_1+\ldots+a_{1n}\ \ x_n & =b_1,\\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ a_{l1}x_1+\ldots+a_{ln}\ \ x_n & =b_l,\\ a_{l+1,1}x_1+\ldots+a_{l+1,n}x_n+x_{n+1} & =b_{l+1},\\ a_{l+2}\ x_1+\ldots+a_{l+2,n}x_n\qquad +x_{n+2} & =b_{l+2},\\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ a_{r1}x_1+\ldots+a_{rn}\ \ x_n\qquad\qquad +x_{n+(r-l)} & =b_r.\end{array}$$

The additional variables appear in the linear form with coefficients which are equal to zero.

If $t=n$, the general linear-programming problem is a priori given in canonical form. In this case, $\bar{n}=n+(r-l)$; $m=r$. (We remind the reader that t is the number of nonnegative variables in the problem (2.1)–(2.4).)

Now, let the mixed restraints be reduced to a system of linear equalities, but let $t<n$, i. e., the nonnegativity requirement does not apply to all the variables. In this case, it is simplest to pass to the canonical form by replacing the variables x_j, which are not restrained by the nonnegativity requirement, by the difference $x_j=x_j'-x_j''$, where $x_j'\geqslant 0$, $x_j''\geqslant 0$.

However, this method will increase the number of variables. The canonical form of a problem may be established differently, by a method which, generally speaking, reduces the number of variables and restraints.

We now apply the linear equations restraining the choice of variables and express the x_j which are not restrained by the nonnegativity requirement in terms of the remaining variables. We then substitute the resulting expressions into the linear form and into the restraints which have not been used in the reduction procedure.

The procedure leads to one of the following cases, depending on the linear form, the linear restraints, and the relationship between the number of restraints and the number of nonnegative variables:

(1) The linear-programming problem is unsolvable.

(2) The total number of variables and restraints is reduced. The linear-programming problem takes on the canonical form which, however, does not guarantee solvability.

5-2. We now give an example of reducing a linear-programming problem to canonical form.

Maximize the linear form

$$L = x_1 + x_2 + cx_3 + x_4$$

subject to the conditions

$$\begin{aligned} ax_1 + x_2 + x_3 - x_4 + 4x_5 &= 4, \\ x_1 + ax_2 + x_3 - 2x_4 - 3x_5 &= 3, \\ x_1 + x_2 - 2x_3 - 3x_4 + 2x_5 &\leq 2, \\ x_1 + x_2 + ax_3 - 4x_4 - 4x_5 &\leq 1, \\ x_4 \geq 0, \quad x_5 &\geq 0. \end{aligned}$$

To reduce the mixed restraints to equalities, we shall introduce two additional variables

$$x_6 \geq 0, \quad x_7 \geq 0.$$

The restraints take on the form

$$\begin{aligned} ax_1 + x_2 + x_3 - x_4 + 4x_5 \qquad &= 4, \\ x_1 + ax_2 + x_3 - 2x_4 - 3x_5 \qquad &= 3, \\ x_1 + x_2 - 2x_3 - 3x_4 + 2x_5 + x_6 &= 2, \\ x_1 + x_2 + ax_3 - 4x_4 - 4x_5 + x_7 &= 1, \\ x_4 \geq 0, \quad x_5 \geq 0, \quad x_6 \geq 0, \quad x_7 &\geq 0. \end{aligned}$$

The variables not restrained by the nonnegativity conditions should now be expressed in terms of the remaining variables. Two cases are possible: (1) three of the four equations are solvable in x_1, x_2, x_3; (2) there are no such equations.

The determinant whose elements are the coefficients of x_1, x_2, and x_3 in the first, second, and fourth equations is equal to $\Delta = (a-1)^2(a+2)$. For $a \neq 1$ and $a \neq -2$, these equations can be used to express x_1, x_2, and x_3 in terms of the nonnegative variables. For example, let $a = 0$. We calculate x_1, x_2, and x_3, and substitute the resulting expressions into the linear form and the third restraint, which has not been used in the procedure.

We obtain

$$\begin{aligned} x_1 &= \qquad \tfrac{5}{2}x_4 + \tfrac{11}{2}x_5 - \tfrac{1}{2}x_7, \\ x_2 &= 1 + \tfrac{3}{2}x_4 - \tfrac{3}{2}x_5 - \tfrac{1}{2}x_7, \\ x_3 &= 3 - \tfrac{1}{2}x_4 - \tfrac{5}{2}x_5 + \tfrac{1}{2}x_7, \end{aligned}$$

$$L = (1+3c) + \frac{1}{2}x_4(10-c) + \frac{1}{2}x_5(8-5c) - \frac{1}{2}x_7(2-c),$$

$$2x_4 + 11x_5 + x_6 - 2x_7 = 7,$$

$$x_4 \geq 0, \quad x_5 \geq 0, \quad x_6 \geq 0, \quad x_7 \geq 0.$$

This is the canonical form of the problem*.

Consider, now, another case, where no three equations of four can be chosen so that the resulting solutions can be used to express those variables which are not bound by the condition of nonnegativity. This case is encountered when, e. g. , $a = -2$. The problem in this case is stated as follows.

Maximize the linear form

$$L = x_1 + x_2 + cx_3 + x_4$$

subject to the conditions

$$\begin{aligned} -2x_1 + x_2 + x_3 - x_4 + 4x_5 \qquad &= 4, \\ x_1 - 2x_2 + x_3 - 2x_4 - 3x_5 \qquad &= 3, \end{aligned}$$

* The constant $(1+3c)$ in the linear form can be omitted since it does not affect the extremum of L.

$$x_1+x_2-2x_3-3x_4+2x_5+x_6=2,$$
$$x_1+x_2-2x_3-4x_4-4x_5+x_7=1,$$
$$x_4\geqslant 0,\quad x_5\geqslant 0,\quad x_6\geqslant 0,\quad x_7\geqslant 0.$$

Here x_1, x_2, and x_3 cannot be expressed in terms of the nonnegative variables, but the first and the second equations can be used to express x_1 and x_2 in terms of the remaining variables. After doing so and substituting the result into the remaining restraints and into the linear form, we obtain

$$\left.\begin{array}{c} L=-7+(2+c)x_3-2x_4+x_5, \\ -6x_4+3x_5+x_6=9, \\ -7x_4-3x_5+x_7=8, \\ x_4\geqslant 0,\quad x_5\geqslant 0,\quad x_6\geqslant 0,\quad x_7\geqslant 0. \end{array}\right\} \tag{5.1}$$

The variable x_3 does not appear in the restraints and thus remains unspecified. If, therefore, the coefficient of x_3 in the linear form is not zero, the linear-programming problem does not have a bounded solution. If, however, $c=-2$, the system (5.1) is the canonical form of a solvable problem.

The above order of computations is independent of the particular example, and may be employed whenever a linear-programming problem is to be reduced to canonical form.

§ 6. Geometrical interpretation of the simplest linear-programming problems

6-1. Linear-programming problems with the number of variables exceeding by two or by three the number of restraints can be interpreted geometrically in the plane or in three-dimensional space. (When we say that the number of variables is greater by two or by three than the number of restraints, we are considering the canonical form of the problem.) Introducing elements of geometry of n-dimensional space, we extend the geometrical interpretation given in this section to problems with any number of variables and restraints. This is called the first geometrical interpretation, and can be applied to linear-programming problems written in any form.

The geometrical interpretation of a problem enables us to discuss in geometrical terms the various methods of solution.

Consider the following particular linear-programming problem.

Maximize the linear form

$$L=c_1x_1+c_2x_2 \tag{6.1}$$

subject to the conditions

$$\left.\begin{array}{c} a_{11}x_1+a_{12}x_2\leqslant b_1, \\ \cdots\cdots\cdots\cdots \\ a_{m1}x_1+a_{m2}x_2\leqslant b_m, \end{array}\right\} \tag{6.2}$$

$$x_1\geqslant 0,\quad x_2\geqslant 0. \tag{6.3}$$

The inequalities (6.3) specify the positive quadrant of the x_1Ox_2-plane. The equation

$$a_{i1}x_1+a_{i2}x_2=b_i, \tag{6.4}$$

associated with the i-th restraint defines a straight line on the x_1Ox_2-plane,

and the inequality

$$a_{i1}x_1 + a_{i2}x_2 \leq b_i \tag{6.5}$$

specifies the half-plane whose direction relative to the line (6.4) is given by the vector $(-a_{i1}, -a_{i2})$, drawn from any point on this line.

The domain of definition of the linear form in the problem (6.1)–(6.3) is, thus, the intersection of the half-planes (6.5) $(i = 1, 2, \ldots, m)$ and the positive quadrant of the x_1Ox_2-plane.

The equality

$$L = c_1x_1 + c_2x_2,$$

for $L = \text{const}$, is the equation of a family of parallel lines. The parameter L of the family is proportional to the distance from the origin to the corresponding line.

In Figure 1.1 we show a case where the domain of definition of the linear form is a polygon. The lines bounding the polygon $OABCD$ in the x_1Ox_2-plane correspond to conditions (6.2) and (6.3), equalities being assumed there. Striation indicates the direction of the half-planes defined by the inequalities (6.2) and (6.3). The direction of the line MN is defined by the vector (c_1, c_2) (the vector (c_1, c_2) is perpendicular to the line MN). The vector (c_1, c_2) also points in the direction of increase of the linear form. The linear-programming problem, i.e., the determination of the coordinates of the point at which the extremum of the linear form (6.1), subject to (6.2) and (6.3), is attained, can be interpreted geometrically for $n = 2$ as follows.

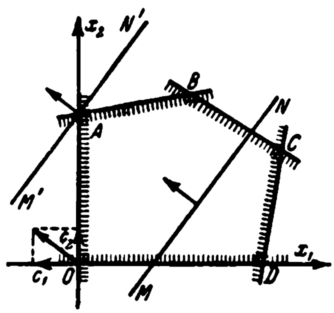


FIGURE 1.1

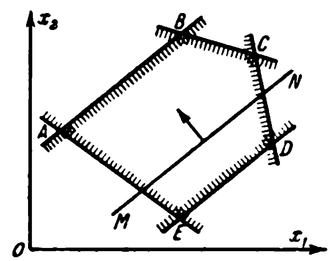


FIGURE 1.2

We draw the line $L = c_1x_1 + c_2x_2$ through the restraint polygon (the domain of definition of the linear form) and translate it parallel to itself in the direction of increasing L, if the linear form is to be maximized, or in the direction of decreasing L, if the linear form is to be minimized. Two cases are possible. In the case shown in Figure 1.1, parallel translation will bring the line into the position $M'N'$, where it has only one common point with the polygon, the common point being the vertex A. This point specifies the unique solution of the linear-programming problem (in Figure 1.1 this is the maximum of the linear form). It may happen, however, that the line MN is parallel to one or to two sides of the polygon. This is shown in Figure 1.2. In this case the extremum is attained at all the points of the corresponding side of the polygon. In Figure 1.2, at all points of the side AB of the polygon $ABCDE$ parallel to the line MN the linear form is maximized, and at all points of the side $ED \parallel MN$, it is minimized. The

linear-programming problem may, thus, have either one or an infinite number of solutions. If two vertices extremize the linear form, all the points of the segment joining these vertices are solutions of the linear-programming problem.

Figure 1.3 corresponds to the case of an unsolvable linear-programming problem: the restraints appear to be contradictory.

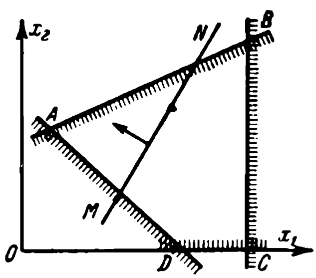


FIGURE 1.3

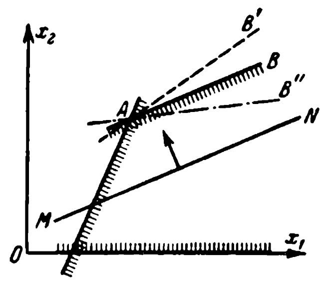


FIGURE 1.4

Figure 1.4 shows an unbounded domain of definition of the linear form. If $AB \parallel MN$, the linear form attains a finite extremum at all points of the ray AB. If the domain of definition of the linear form is modified by rotating the ray AB about the point A, we have two cases: either the linear form is not bounded at the feasible values of the variables, or it is maximized at a single point. The first case is depicted by the ray AB'—the dashed line (Figure 1.4). The second case is depicted by AB''— the dash-dot line (Figure 1.4).

6-2. Until now we have considered the geometrical interpretation of a linear-programming problem in two variables and m inequality restraints of the form (6.2). If this problem is reduced to canonical form, the number of variables of the new problem will be $n = m+2$, and the number of restraints (equalities) will be m. We thus have $n-m=2$.

It can easily be seen that the geometrical interpretation of linear-programming problems given in 6-1 applies only when $n-m=2$, n and m being arbitrary. We express all the variables in terms of only two of them, say x_1 and x_2, and rewrite the linear form and the restraints as follows:

$$L = c_1'x_1 + c_2'x_2,$$
$$x_s = a_{1s}'x_1 + a_{2s}'x_2 - a_s', \qquad s = 3, 4, \ldots, n,$$
$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n.$$

The lines $x_s = 0$ $(a_{1s}'x_1 + a_{2s}'x_2 = a_s')$ and the axes define a convex polygon (a convex polygonal set) in the $x_1 O x_2$-plane. The conditions $x_s \geqslant 0$ specify the direction of striation of the polygon sides. We have thus specified the range of all the possible pairs (x_1, x_2) amongst which the points maximizing (or minimizing) the linear form must be chosen. The coefficients of the linear form define a family of parallel lines and the direction in which L increases. We choose from this family a line intercepting the polygon and translate it in the direction of increasing (decreasing) linear form, until it has no common points with the polygon. The limiting position of the line specifies the maximum (minimum) of the linear form.

As illustrative is the geometrical interpretation of the linear-programming problem for three variables, when the restraints are inequalities, or

when the number of variables exceeds by three the number of restraints ($n-m=3$) and the problem is given in canonical form. The restraints define in space a convex polyhedron (a convex polyhedral set). The linear-form coefficients specify a family of parallel planes H and the direction in which L increases. To solve a linear-programming problem, the plane H intercepting the polyhedron should be translated in the direction of increasing linear form, if maximization is required, or in the direction of decreasing L, if minimization is required, until it has no common points with the polyhedron. The limiting position of the plane specifies the solution of the problem.

Reasoning geometrically shows, as before, that the form is extremized at the vertices of the polyhedron. If the form is extremized at more than one point, it is extremized over the entire edge or face of the polyhedron parallel to the plane defined by the linear-form coefficients.

In the following we shall see that all the features of linear-programming problems and their methods of solution can be explained within the framework of this geometrical interpretation.

6-3. We now give another geometrical interpretation of linear-programming problems. The s e c o n d g e o m e t r i c a l i n t e r p r e t a t i o n illustrates methods of solution of canonical problems. We shall consider problems with two restraints and any number of variables. These problems can be visualized in three-dimensional space. Suppose it is required to maximize the linear form

$$L=c_1x_1+c_2x_2+\dots+c_nx_n \tag{6.6}$$

subject to the conditions

$$\left.\begin{array}{l} a_{11}x_1+a_{12}x_2+\dots+a_{1n}x_n=b_1, \\ a_{21}x_1+a_{22}x_2+\dots+a_{2n}x_n=b_2, \end{array}\right\} \tag{6.7}$$
$$x_j\geqslant 0, \quad j=1,\ 2,\ \dots,\ n$$

We introduce the new variables

$$\left.\begin{array}{l} u_1=a_{11}x_1+\dots+a_{1n}x_n, \\ u_2=a_{21}x_1+\dots+a_{2n}x_n, \\ u_3=c_1x_1\ +\dots+c_nx_n \end{array}\right\} \tag{6.8}$$

Relationships (6.8) define a transformation of the n-dimensional space of the variables $x_1,\ x_2,\ \dots,\ x_n$ into the three-dimensional space of the variables $u_1,\ u_2,\ u_3$. The positive semiaxes $Ox_1,\ Ox_2,\ \dots,\ Ox_n$ of the n-dimensional space are transformed into rays issuing from the origin. Indeed, the coordinates of the positive semiaxis, Ox_1, satisfy the conditions $x_2=x_3=\dots=$ $=x_n=0$. The image of the Ox_1-axis in the $u_1,\ u_2,\ u_3$-space is therefore defined by the relationships

$$u_1=a_{11}x_1, \quad u_2=a_{21}x_1, \quad u_3=c_1x_1, \quad 0\leqslant x_1<\infty. \tag{6.9}$$

Hence, the image of the Ox_1-semiaxis is a ray in the three-dimensional space of $u_1,\ u_2,\ u_3$. Similarly, the positive semiaxis, Ox_k,

$$x_1=\dots=x_{k-1}=x_{k+1}=\dots=x_n=0,\ 0\leqslant x_k<\infty,$$

is transformed into the ray

$$u_1=a_{1k}x_k,$$
$$u_2=a_{2k}x_k, \quad u_3=c_kx_k,$$
$$0\leqslant x_k<\infty.$$

Figure 1.5 shows, in the coordinates u_1, u_2, u_3, the rays Λ_j corresponding to the positive semiaxes in the $x_1, \ldots, x_n$-space. All the rays Λ_j define a convex polyhedral cone K, the image of the orthant* $x_j \geqslant 0$ $(j=1, 2, \ldots, n)$ in the x-space. The orthant $x_j \geqslant 0$ is a convex set. It can easily be shown that a linear transformation transforms a convex set into a convex set. Hence the convexity of the polyhedral cone.

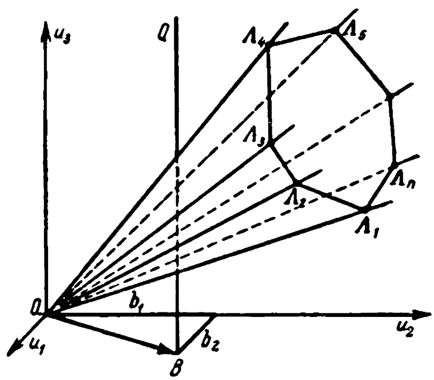


FIGURE 1.5

Let $\bar{A}_j$ be a three-dimensional (and, generally, $(m+1)$-dimensional) vector with components a_{1j}, a_{2j}, c_j (in general, with components $a_{1j}, a_{2j}, \ldots, a_{mj}, c_j$),

$$\bar{A}_j = (a_{1j}, a_{2j}, c_j)^T \quad (\bar{A}_j = (a_{1j}, a_{2j}, \ldots, a_{mj}, c_j)^T).$$

The first two (m) components of the vector $\bar{A}_j$ coincide with the components of the restraint vector, and the third ($(m+1)$-th) component is equal to the corresponding linear-form coefficient. We shall refer to vector $\bar{A}_j$ as the augmented restraint vector. Augmented restraint vectors define the direction of the Λ_j rays, the images of the positive semiaxes in the $x_1, x_2, \ldots, x_n$-space. It follows from (6.8) that the positive orthant in the x-space is transformed in the three-dimensional space into a convex cone defined by the augmented restraint vectors

$$\bar{A}_1, \bar{A}_2, \ldots, \bar{A}_n.$$

6-4. To study the features of linear-programming problems, we should consider not the image of the positive orthant, but rather the image of the domain of definition of the particular problem in question. It is, therefore, expedient to consider, together with the rays Λ_j (the images of the positive semiaxes), also the line Q defined by

$$\begin{aligned} u_1 &= b_1, \\ u_2 &= b_2, \\ u_3 &= q \ (-\infty < q < \infty), \end{aligned}$$

where b_i are the components of the constraint vector B.

The line Q is parallel to the Ou_3-axis, and its projection onto the (u_1, u_2)-plane is the constraint vector B. In Figure 1.5, Q is vertical. Its projection onto the u_1Ou_2-plane has the components (b_1, b_2).

* A positive orthant is generally defined as the n-dimensional analog of the positive octant, i.e., the set of vectors X with nonnegative coordinates.

We shall show that the image of the domain of definition of a linear-programming problem in the u_1, u_2, u_3-space is the intersection Q_K of the line Q and the convex polyhedral cone K. Indeed, for any point $u=(u_1, u_2, u_3)\in Q_K$, we have

$$u_1 = a_{11}x_1 + \ldots + a_{1n}x_n = b_1,$$
$$u_2 = a_{21}x_1 + \ldots + a_{2n}x_n = b_2,$$

since Q_K is a subset of Q. Moreover, for all the points $X\in Q_X$ which are the images of the points $u\in Q_K$

$$x_1 \geqslant 0, \quad x_2 \geqslant 0, \quad \ldots, \quad x_n \geqslant 0,$$

since the corresponding points u belong to the polyhedral cone, which is the image of the positive orthant in the x-space. To each point of the intersection of the line Q and the cone K there correspond certain points from the domain of definition of the linear form. The converse is obviously also true: each point of the domain of definition of the problem is mapped onto a point of the set Q_K, which is the intersection of the polyhedral cone and the line Q. The set Q_K is a segment, a ray or a line.

To each point $X\in Q_X$ there corresponds a certain $u_3=q$. But

$$u_3 = c_1x_1 + \ldots + c_nx_n$$

specifies the value of the linear form L. Hence the parameter q of the line Q corresponds to the magnitude of the linear form. If the line Q contains inside points of the convex polyhedral cone, the boundary of the cone has only two points of intersection with this line. One of these corresponds to the maximum of the linear form, the other to the minimum. If the line Q is tangent to the cone, i. e., if it belongs to one of its faces, the upper point of contact specifies the maximum of the linear form, and the lower its minimum.

The linear-programming problem, i. e., the problem of extremizing the linear form (6.6) subject to (6.7), is reduced to the determination of the extremum (upper in the case of maximization and lower in the case of minimization) point of intersection of the line Q with the convex polyhedral cone.

If Q does not intersect the convex polyhedral cone defined by the augmented restraint vectors $\bar{A}_1$, $\bar{A}_2$, ..., $\bar{A}_n$, the domain of definition of the linear-programming problem is empty. In this case the problem is obviously unsolvable. If the intersection of Q and the convex polyhedral cone is not bounded (from above for a maximization problem, and from below for a minimization problem), the problem in question is unsolvable because the linear form is not bounded in the domain of its definition.

In Chapter 4 the two geometrical interpretations are generalized for linear-programming problems with any number of variables and restraints.

§ 7. Economic interpretation of linear-programming problems

7-1. Economic problems constitute the main source of linear-programming problems. In fact, linear programming arose as a tool for filling the requirements of economics. It is, therefore, advisable to supplement the geometrical interpretation of the linear-programming problem by the corresponding economic interpretation.

Consider the scheduling of operations of an enterprise or of a group of enterprises producing some homogeneous commodity. The production of the commodity requires several types of raw materials, certain machine-tools, certain man-hour potentials of laborers of various qualifications, power, fuel, transport, etc. We shall combine all these aspects under the heading of production factors. Let the number of factors specifying the production technology be m. Each of these is, as a rule, available in the enterprise in limited quantities.

Each feasible set of production factors ensures a finite production output in unit time. We shall use the term "mode of production" to designate a system of $m+1$ numbers (an $(m+1)$-dimensional vector) specifying the requirements of each production factor in unit time and the corresponding output.

In each problem several modes of production can be indicated, each utilizing the production factors in manufacturing the commodity differently. The following scheduling procedure of an enterprise is feasible.

First prepare (or assume as known) several modes of production, i. e., several sets specifying the requirements of each production factor in unit time and the corresponding output of the commodity. The consumption of production factors of different categories and the output of the commodity depends on the time during which the enterprise operates according to each of the feasible modes of production.

The scheduling problem is, thus, to determine the time to be assigned to each individual mode of production.

The optimal schedule is that program in which the consumption of each production factor does not exceed its admissible value, and the output of the enterprise attains the highest level. Obviously, the optimal plan is the one where the greatest number of production modes are subjected to analysis, and processing.

During the formulation of the scheduling problem (in the previous sense) the following assumptions, which appear quite natural in the general context, should be made.

If one mode of production ensures an output of c_1 units of a particular commodity in unit time and another mode, c_2 units, then by scheduling the enterprise to operate for x_1 time units according to the first mode and x_2 time units according to the second mode we ensure an output of $c_1x_1+c_2x_2$ units of the commodity. There is, obviously, no need to assume that the time intervals x_1 and x_2 are consecutive. The enterprise may be simultaneously scheduled to operate according to different modes of production, with some modes being replaced by other modes at definite intervals. The time intervals x_1 and x_2 should thus be interpreted as the specific weights of the particular production modes in the production program of the enterprise.

The time of switching over from one mode of production to another, i.e., the idling time, is not taken into consideration here.

7-2. We now translate the scheduling problem into the language of mathematics.

Let c_j be the number of units of the commodity produced by the plant in unit time of operation according to the j-th production mode, where the consumption of the first production factor is a_{1j}, of the second production factor a_{2j}, ..., of the i-th — a_{ij}, ..., and of the m-th production factor a_{mj}. Let, moreover, the quantities $b_1, b_2, \ldots, b_m$ denote the available resources of the various production factors. The quantities b_i limit the permissible consumption.

Assume that in our scheduling we have at our disposal n modes of production. Let x_j be the time interval during which the plant is scheduled to operate according to the j-th mode. The total output of the plant is then determined by

$$L=c_1x_1+c_2x_2+\ldots+c_nx_n. \tag{7.1}$$

The restraints imposed on the consumption of the individual production factors are written in the form

$$\left.\begin{array}{l} a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n\leqslant b_1, \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ a_{m1}x_1+a_{m2}x_2+\ldots+a_{mn}x_n\leqslant b_m. \end{array}\right\} \tag{7.2}$$

The time intervals are, obviously, nonnegative, i. e.,

$$x_j\geqslant 0, \quad j=1,\ 2,\ \ldots,\ n. \tag{7.3}$$

Observe that the scheduling problem has been reduced to a linear-programming problem. The choice of the program is the determination of the set of numbers $x_1, x_2, \ldots, x_n$, for which conditions (7.2) and (7.3) are satisfied and the form (7.1) is maximized.

The terminology of linear-programming problems used in economics is also convenient when treating other problems described by the same formal apparatus. The economic interpretation of the linear-programming problem will be useful later on for an intuitive substantiation of the various concepts introduced in algorithms. The vectors

$$A_j=(a_{1j},\ \ldots,\ a_{mj})^T,\ B=(b_1,\ \ldots,\ b_m)^T$$

were previously termed the restraint vector and the constraint vector, respectively. In economic interpretation, the components of the restraint vector specify the demand in individual production factors in unit time for the corresponding mode of production, and the components of the constraint vectors give the supply of the individual factors limiting the actual demand.

In terms of the production problem, the vectors A_j $(j=1,\ 2,\ \ldots,\ n)$ and the vector B should, naturally, be interpreted as the demand vector and the supply vector, respectively.

The j-th mode of production is defined by the vector $\bar{A}_j$, whose first m components coincide with the components of the restraint vector A_j, and whose last component is c_j, i. e., the output in unit time. According to the definition introduced in § 6, $\bar{A}_j$ will be considered as an augmentation of the vector A_j. Augmented vectors characterize entirely the modes of production, both specifying the inputs and the outputs.

A set of numbers $x_1, x_2, \ldots, x_n$ satisfying the restraints of a linear-programming problem was called (see § 2) a program. This term, which is self-explanatory in problems of production scheduling, is retained for all linear-programming problems regardless of their context. The same applies to the term optimal program and to certain additional terms which will be introduced in the following.

7-3. The problem stated here in terms of production scheduling may be modified to fit the requirements of the static model of national-economy planning.

Consider a model of interaction between various branches of industry in an economic region. Let the economy of the region be shaped by n industrial branches. For simplicity we assume that the region is economically

selfcontained. The production schedule for the output commodity* of each branch is assumed to be known. Part of the output is exported to other regions and part is locally consumed. All the branches of industry are interdependent and normal functioning of each branch is impossible unless an orderly supply of production means from other branches is maintained. The production program for individual branches of the economy and the interaction between the various branches should ensure minimum labor demand (in man-hours and in machine facilities) for the production of output commodities in quotas specified and according to a preset schedule.

We now translate this problem of planning the interaction between the branches of industry in a selfcontained economic region into formal language. In order to be specific, we shall assume that an annual plan is drawn. We introduce the following notations.

Let x_i be the total output of the i-th branch during the period specified; x_{ij} the total amount of materials supplied by the i-th branch to the j-th branch in one year; and y_i the annual quota of output commodity of the i-th branch.

The parameters x_{ij} and y_i, like x_i, are measured in natural units; y_i is specified by government economic programs which, in particular, take into consideration the requirements of the region.

The problem of planning regional economy under a static model consists of drawing quotas for the various branches of industry which would fill the demand in products of any branch in all other dependent branches, and would also satisfy the state quotas for the output commodity of each branch.

Applying mathematical notations, we represent the requirements of the problem as a system of inequalities

$$x_i \geqslant \sum_{j=1}^{n} x_{ij} + y_i. \tag{7.4}$$

The ratio $\frac{x_{ij}}{x_j} = a_{ij}$ specifies the amount of the commodity, produced by the i-th branch, required for producing a unit of commodity released by the j-the branch. In these new notations, conditions (7.4) are written in the form

$$x_i - \sum_{j=1}^{n} a_{ij} x_j \geqslant y_i, \tag{7.5}$$

or, in matrix form,

$$(I - A) X \geqslant Y, \tag{7.6}$$

where I is the unit matrix of order n; $A = \| a_{ij} \|_n$ is the interaction matrix; and X and Y are n-dimensional vectors with components x_j and y_i, respectively. The vector X is the production vector, and Y is the output commodity vector.

The matrix $I - A$ is generally referred to in economics as the Leontieff matrix. Given the vector X, W. Leontieff calculated the vector Y for some time in advance, proceeding from the values of a_{ij} established on the basis of previous experience.

Until now we have assumed, implicitly, that there exists only one technological process for utilizing the products of various regional branches of the economy in manufacturing the commodity of any given branch. It seems, however, that alternative interaction models also exist /65/.

* We remind the reader that an output commodity is a commodity which is not used as raw material in some other production processes (consumer items, goods for export, military equipment, etc.).

Normal output of each branch can generally be ensured for various relationships with the other branches of the economic region. A unit output of each branch may be produced for different patterns of supply from the other branches. The labor requirements for producing one unit of a commodity obviously depend on the pattern chosen.

Let each branch function under l different production modes. Each technological process for producing the commodity of the j-th branch is characterized by the set of numbers $a_{ij}^{(s)}$ $(i=1, 2, \ldots, n)$, and $c_j^{(s)}$. The matrix $A^{(s)}=\|a_{ij}^{(s)}\|_n$ defines the industrial interdependence for the s-th mode of production. The vector $C^{(s)}=(c_1^{(s)}, \ldots, c_n^{(s)})$ characterizes the labor requirements for unit output of each branch of the economy scheduled for the s-th production mode.

As the technologically feasible modes of production in each branch become more numerous, i. e., the range of values assumed by the index s increases, the scope of feasible programs becomes wider and the labor requirements corresponding to the optimal plan drop closer to the minimum.

The best program is determined by the vectors X_s minimizing the linear form

$$L=\sum_{s=1}^{l} C^{(s)}X_s \tag{7.7}$$

subject to the conditions

$$\sum_{s=1}^{l} (I-A^{(s)})\,X_s \geqslant Y, \tag{7.8}$$

$$X_s \geqslant 0. \tag{7.9}$$

Here we have assumed that all the branches of the economy have the same number of technologically feasible production modes.

If, however, the j-th branch has l_j modes of production, the problem is stated as follows.

Determine the variables x_{sj} minimizing the linear form

$$L=\sum_{j=1}^{n}\sum_{s=1}^{l_j} c_j^{(s)}x_{sj} \tag{7.10}$$

subject to the conditions

$$\sum_{s=1}^{l_i} x_{si}-\sum_{j=1}^{n}\sum_{s=1}^{l_j} a_{ij}^{(s)}x_{sj}\geqslant y_i, \tag{7.11}$$

$$x_{sj}\geqslant 0, \tag{7.12}$$

$$i, j=1, 2, \ldots, n; \quad s=1, 2, \ldots, l_j.$$

Here $a_{ij}^{(s)}x_{sj}$ is the output of the i-th branch required for producing x_{sj} units of the commodity of the j-th branch, according to the s-th technological method; $\sum\limits_{s=1}^{l_j} a_{ij}^{(s)}x_{sj}$ is the output of the i-th branch required for filling the quota of the j-th branch; $\sum\limits_{s=1}^{l_i} x_{si}$ is the total output of the i-th branch; $c_j^{(s)}x_{sj}$ are the labor requirements for manufacturing x_{sj} units of the commodity of the j-th branch by the s-th technological method; and $\sum\limits_{j=1}^{n}\sum\limits_{s=1}^{l_j} c_j^{(s)}x_{sj}$ is the total labor requirement for filling all the regional quotas by all the feasible technological methods.

It can be shown (see Exercise 4) that given a single technological mode

of production $(s=1)$, the optimal program reduces the inequality (7.8) to an equality. The program balancing the input and the output, i. e., the program X satisfying the equation $(I-A)X=Y$, is in this case the optimal program for any demand vector $C \geqslant 0$ /83/. In the general case, however, the optimal plan does not reduce inequalities (7.8) or (7.11) to equalities.

Observe that the problem of national-economy planning, like the problem of industrial scheduling, is a general linear-programming problem. We have already seen that certain theoretical concepts and some principles of method design in linear programming have a manifest economic background. By formulating the general postulates of linear programming in terms of economic concepts we acquire the intuition for assessing new trends in the development of this discipline.

EXERCISES TO CHAPTER 1

1. Proceeding from the arguments of 3-1, find the measure function and the restraints for an allocation problem dealing with the supply of an inhomogeneous product to n destinations from m production origins.

2. Formulate and reduce to a linear-programming problem the following problem of allocation for a sowing area.

Sowing areas of collective farms in a certain region are $s_1, s_2, \ldots, s_n$, respectively. According to a master plan, the crops should be produced in the region in the proportion $p_1:p_2:\ldots:p_m$. The expected output of an i-th crop from the field of the j-th farm is a_{ij} units from 1 acre. Determine the fraction of sowing area to be allocated to each crop in each farm to ensure maximum output with the prescribed crop ratio.

3. Solve graphically a scheduling problem (see 7-2) for the case when the supply vector $B=(50, 20, 60, 90)^T$, and the augmented demand vectors of two production modes are, respectively,

$$\bar{A}_1=(2, 2, 5, 1, 1)^T,$$
$$\bar{A}_2=(5, 1, 6, 10, 1)^T.$$

4. Prove that in a model of national-economy planning (see 7-3) limited by a single technological factor $(s=1)$, the optimal plan reduces the inequality (7.8) to an equality.

5. Find the canonical form of the problem maximizing the linear form

$$L=x_1-x_2$$

subject to the conditions

$$\begin{aligned} -2x_1+ax_2 &\leqslant 2, \\ x_1-bx_2 &\leqslant 3, \\ x_1+x_2 &\geqslant c, \\ dx_1+\frac{x_2}{2} &\leqslant 1, \\ x_1 &\geqslant 0. \end{aligned}$$

Analyze the different cases depending on the value of the constants a, b, c, and d.

6. Applying the first geometrical interpretation of the linear-programming problem, determine the constants in Example 5 for which

(a) the solution is unique;

(b) the restraints are inconsistent;

(c) the domain of definition of the linear form is not bounded, but the problem is solvable;

(d) the linear form is not bounded in its domain of definition;

(e) the problem has an infinite number of solutions.

7. Proceeding from the geometrical interpretation of the linear-programming problem, determine the range of the linear form

$$L=x_1-x_2,$$

if the variables are limited by the conditions

$$\begin{aligned} x_1+3x_2+x_3+x_4+2x_5&=6, \\ -2x_1+x_2+3x_3-4x_4+x_5&=2, \\ 5x_1-2x_2+x_3+2x_4-3x_5&=-4, \\ x_j\geqslant 0, \quad j=1,\ 2,\ 3,\ 4,\ 5. \end{aligned}$$

8. Applying the second geometrical interpretation of the linear-programming problem, indicate the range of the parameters a, b, and c in which the following problem is solvable:

Maximize the linear form

$$L=x_1+3x_2+cx_3-x_4$$

subject to the conditions

$$\begin{aligned} ax_1-2x_2+x_3-3x_4&=5, \\ x_2+2x_3+x_4&=b, \\ x_j\geqslant 0, \quad j=1,\ 2,\ 3,\ 4. \end{aligned}$$

Chapter 2

. CONVEX POLYHEDRAL SETS AND LINEAR PROGRAMMING

The general linear-programming problem is concerned with determining the extremum (minimum or maximum) of a linear function whose variables are restrained by several linear relationships (equalities or inequalities). The set of points defined in the space of the variables by these restraints is a many-dimensional analog of the two-dimensional convex polygonal domain whose boundary is a polygonal line with a finite number of sides. These sets are generally called c o n v e x p o l y h e d r a l s e t s.

The linear-programming problem thus deals with the analysis of a linear function defined on some convex set. Hence the close relationship between linear programming and the theory of convex polyhedral sets.

In this chapter we present some elements of this theory which are essential for substantiating most of the results of linear programming (§ 1 – § 3). Although our approach to the theory of convex polyhedral sets differs considerably from other known methods used in discussing this theory (see /23, 24/, and also /107/), we believe that it is more natural and illustrative.

In § 4 the results of the previous sections are used to establish several important theorems of linear programming. The concluding § 5 gives two different geometrical interpretations of linear-programming problems which, we hope, will be conducive to a better understanding of the subject.

§ 1. Convex polyhedral sets

1-1. Consider all the feasible programs of an arbitrary linear-programming problem in n variables. In other words, consider a set of points (vectors) $X=(x_1, \ldots, x_n)$ defined by an arbitrary system of linear restraints (equalities and inequalities):

$$(D_i, X)=\sum_{j=1}^{n} d_{ij}x_j \leqslant d_i, \qquad i=1, 2, \ldots, s; \tag{1.1}$$

$$(D_i, X)=\sum_{j=1}^{n} d_{ij}x_j = d_i, \qquad i=s+1, s+2, \ldots, s+t. \tag{1.2}$$

We do not write the conditions of the form $x_j \geqslant 0$ separately, as is usually done in formulating the restraints of a linear-programming problem. These conditions (if specified) are included in the general system of inequalities (1.1). Let M be the set of points X satisfying (1.1) and (1.2).

We shall assume that the system (1.1), (1.2) is consitsent, i. e., the set M is not empty. In this section we propose to establish several simple, and yet important properties of the set M which consists of all the feasible programs of the general linear-programming problem.

Theorem 1.1. The set M is a convex closed set.*

Proof. 1. Let X_1, X_2 be any two points of M; let $0 \leqslant \alpha \leqslant 1$. In this case the point $X = \alpha X_1 + (1-\alpha) X_2 \in M$. Indeed, $(D_i, X) = \alpha(D_i, X_1) + (1-\alpha)(D_i, X_2)$. Hence

$$(D_i, X) \leqslant \alpha d_i + (1-\alpha) d_i = d_i, \qquad i = 1, 2, \ldots, s;$$
$$(D_i, X) = \alpha d_i + (1-\alpha) d_i = d_i, \qquad i = s+1, s+2, \ldots, s+t,$$

i. e., $X \in M$.

By definition the set M is, therefore, convex.

2. Let X_k, $k = 1, 2, \ldots$ be an arbitrary sequence of points in M converging to the point X. Since the scalar product of vectors is a continuous function, we have

$$\lim_{k \to \infty} (D_i, X_k) = (D_i, X) \tag{1.3}$$

for any i. By assumption,

$$(D_i, X_k) \leqslant d_i, \qquad i = 1, 2, \ldots, s,$$
$$(D_i, X_k) = d_i, \qquad i = s+1, \ s+2, \ldots, s+t, \quad (k = 1, 2, \ldots).$$

In these relationships passing to the limit as $k \to \infty$ and applying (1.3), we observe that the point X satisfies (1.1) and (1.2), i. e., belongs to the set M. The set M thus contains all its cluster points and as such is a closed set. This completes the proof.

The structure of (1.1), (1.2) shows that the set M is actually the intersection of the half-spaces $(D_i, X) \leqslant d_i$, $i = 1, 2, \ldots, s$ with the hyperplanes $(D_i, X) = d_i$, $i = s+1, s+2, \ldots, s+t$, in the n-dimensional space of the points (vectors) $X = (x_1, x_2, \ldots, x_n)$. The definition of a hyperplane and a half-space for any finite-dimensional space is given in the Appendix, 1-7.

The set comprising the intersection of a finite number of half-spaces and hyperplanes (provided this intersection is not empty) will be called a convex polyhedral set. M is thus a convex polyhedral set.

The preceding definition indicates that any convex polyhedral set can be defined by a consistent system of restraints of the form (1.1), (1.2).

Since in the following we shall deal only with convex polyhedral sets, the adjective convex is dropped wherever no confusion can arise. The system (1.1), (1.2) generating the convex polyhedral set M is sometimes called the restraint system of the set.

A convex polyhedron is defined as a bounded convex polyhedral set. If the set M is bounded, (1.1), (1.2) define a convex polyhedron in an n-dimensional space.

According to the preceding definitions, all the feasible programs of a linear-programming problem can be described as the polyhedral restraint set (or the restraint polyhedron) of the problem.

1-2. We now establish the relation between the dimensionality of the polyhedral set M (see Appendix, 3-4) and the properties of the restraints (1.1), (1.2) generating this set. First we give some definitions.

The i-th constraint of the set (1.1), (1.2) will be called a rigid constraint of the polyhedral set M if any point of M satisfies the corresponding equality, i. e., for $X \in M$, $(D_i, X) = d_i$.

Any of the conditions (1.2) is obviously a rigid constraint imposed on M. Some of the conditions (1.1) may also prove to be rigid constraints.

* For the definition of a convex set and a closed set, see Appendix, § 3.

For example, let us consider a polyhedral set M' defined by the inequalities

$$\left.\begin{aligned} x_1+x_2&\leqslant 1,\\ 2x_1+x_2&\leqslant 2,\\ -3x_1-2x_2&\leqslant -3,\\ x_3&\leqslant 1.\end{aligned}\right\} \tag{1.4}$$

Let $X=(x_1, x_2, x_3)\in M'$. Adding the first two restraints in (1.4) and comparing the result with the right-hand side of the third inequality of the set, we obtain

$$-3x_1-2x_2=-3.$$

The third inequality of (1.4) thus imposes a rigid constraint on the polyhedral set M'.

The i-th constraint of the system defining the polyhedral set M will be called a **nonrigid constraint** of this set if a point $X\in M$ exists such that

$$(D_i, X)<d_i.$$

The system of constraints defining a polyhedral set can thus be divided into two subsystems, one comprising rigid, and the other nonrigid constraints of the given set. Nonrigid constraints are, obviously, provided only by the inequalities. In the preceding example, the fourth inequality imposes a nonrigid constraint on M', since the point $(x_1', x_2', x_3')=(1,0,0)\in M'$ and $x_3'=0<1$. It can be easily verified that the first two inequalities of (1.4) are actually rigid constraints of M'.

A system of linear constraints comprising conditions of the form (1.1) and (1.2) is said to be **linearly independent** if the corresponding system of vectors D_i is linearly independent. The **rank** of a system of linear constraints is the rank of the matrix constructed from the vectors D_i associated with the given system.

We now proceed with the statement and the proof of the dimensionality theorem of polyhedral sets.

Theorem 1.2. *The dimensionality ϱ of a polyhedral set M is given by*

$$\varrho=n-\sigma, \tag{1.5}$$

where σ is the rank of the system of rigid constraints of this set.

Proof. Let the set of subscripts of the rigid and the nonrigid constraints imposed on M be denoted by E_1 and E_2, respectively. By assumption, the convex polyhedral set M is the intersection of the hyperplanes

$$(D_i, X)=d_i, \quad i\in E_1, \tag{1.6}$$

σ of which are linearly independent. (A system of hyperplanes is said to be **linearly independent**, if the direction vectors of these hyperplanes are linearly independent.) Hence, according to the definition of dimensionality of a convex set

$$\varrho\leqslant n-\sigma. \tag{1.7}$$

Consider a system of linear homogeneous equations

$$(D_i, X)=0, \quad i\in E_1. \tag{1.8}$$

Since the rank of the system of vectors D_i, $i\in E_1$, is σ, there exist $n-\sigma$ linearly independent vectors X_l, $l=1, 2, \ldots, n-\sigma$, satisfying equations (1.8).

Moreover, for any $i \in E_2$ there exists a vector Y_i satisfying (1.6) so that

$$(D_i, Y_i) < d_i.$$

Taking

$$Y_0 = \frac{1}{q} \sum_{i=1}^{q} Y_i,$$

where q is the number of nonrigid constraints imposed on M, we have

$$(D_i, Y_0) < d_i \tag{1.9}$$

for all $i \in E_2$. Since the set M is convex, $Y_0 \in M$. It follows from (1.9) that for any $i = 1, 2, \ldots, n-\sigma$ and sufficiently small ε

$$Y_0 + \varepsilon X_i \in M.$$

Now, assume that the polyhedral set M is the intersection of some hyperplanes

$$(\Lambda_\alpha, X) = \lambda_\alpha, \qquad \alpha = 1, 2, \ldots, l.$$

Since

$$Y_0 \in M \text{ and } Y_0 + \varepsilon X_i \in M,$$

$$(\Lambda_\alpha, X_i) = \frac{1}{\varepsilon} [(\Lambda_\alpha, Y_0 + \varepsilon X_i) - (\Lambda_\alpha, Y_0)] = \frac{1}{\varepsilon} (\lambda_\alpha - \lambda_\alpha) = 0.$$

The system of homogeneous equations

$$(\Lambda_\alpha, X) = 0, \qquad \alpha = 1, 2, \ldots, l$$

thus has $n-\sigma$ linearly independent solutions X_i. Hence, among the vectors Λ_α, $\alpha = 1, 2, \ldots, l$ there are not more than σ linearly independent vectors. Thus, we have established that any system of hyperplanes whose intersection contains the convex polyhedral set M comprises not more than σ linearly independent hyperplanes. Therefore,

$$\varrho \geqslant n - \sigma.$$

Comparing this result with the inequality (1.6), we obtain (1.5). This completes the proof.

Consider now the polyhedral set M' defined by (1.4). The first three inequalities in (1.4) impose rigid constraints on M'. The first two constraints are linearly independent, while the third is a linear combination of them. Hence, in this particular case, $\sigma = 2$. The dimensionality of the polyhedral set M' is therefore

$$\varrho = n - \sigma = 3 - 2 = 1.$$

The system of constraints defining a polyhedral set may be reduced to an equivalent form with each inequality imposing a nonrigid constraint and all the equalities being linearly independent. To this end it suffices to replace, in each rigid inequality constraint, the inequality by an equality. Then, those equations of the system which are a linear combination of others, should be eliminated. The system thus obtained is equivalent to the original system in the sense that both define the same polyhedral set. Assuming that these manipulations have been carried out, and that (1.1), (1.2) is the resulting system, then, according to Theorem 1.2, the dimensionality ϱ of the set M is

$$\varrho = n - t. \tag{1.10}$$

1-3. In what follows we shall use the term "face" of an arbitrary

polyhedral set. A subset G of a polyhedral set M is called a q-dimensional face of M if

(a) G is q-dimensional;

(b) from $X=\alpha X'+(1-\alpha)X''\in G$, $0<\alpha<1$ and $X', X''\in M$ it follows that $X', X''\in G$.

It is easily observed that for $q=0$ this definition reduces to the definition of an extremum point of the set M. Extremum points of a polyhedral set are, naturally, referred to as vertices of this set. Thus, the zero-dimensional face of M and the vertex of M are equivalent concepts.

The edge of a polyhedral set is defined as any one-dimensional face of the set.

The face of a polyhedral set M can also be defined in terms of (1.1), (1.2), i. e., a q-dimensional face of the set M is any q-dimensional polyhedral set generated by restraints obtained from (1.1), (1.2) when some of the inequalities are replaced by equalities. Observe that by Theorem 1.2 the number of linearly independent rigid constraints of a q-dimensional face is $n-q$.

Theorem 1.3. The two definitions given above of a q-dimensional face of a polyhedral set are equivalent.

Proof. 1. Let G be a q-dimensional face of a polyhedral set M in the sense of the first definition. Let E_G be the set of subscripts i so that

$$(D_i, X)=d_i$$

for all $X\in G$. Obviously $E_G\supseteq E_1$, where E_1 is the set of subscripts of the rigid constraints of M. Let G' be a polyhedral set whose restraints are obtained from (1.1), (1.2) when the inequalities for $i\in E_G$ are replaced by equalities (this obviously refers only to those $i\in E_G$ associated with restraints of the form (1.1)). We shall show that $G=G'$. Let X be a point in G'; X_0 is a point in G such that

$$(D_i, X_0)<d_i, \quad i\notin E_G, \tag{1.11}$$
$$(D_i, X_0)=d_i, \quad i\in E_G. \tag{1.12}$$

The existence of this point follows from the definition of the set E_G.

Let $X'=X_0+\varepsilon(X_0-X)$, $\varepsilon>0$. Obviously, X' satisfies the equalities (1.12) for any ε and the inequalities (1.11) for sufficiently small ε. Hence, for sufficiently small ε, $X'\in G'$. Thus, $X, X'\in G'\subseteq M$, $X_0\in G$, where

$$X_0=\frac{1}{1+\varepsilon}X'+\frac{\varepsilon}{1+\varepsilon}X.$$

In this case it follows from the definition of a face of a polyhedral set (condition (b)) that

$$X', X\in G.$$

An arbitrary point $X\in G'$ thus belongs to G, i. e., $G'\subseteq G$. The converse, $G\subseteq G'$, follows immediately from the definition of G'. Thus G coincides with G' and, correspondingly, satisfies the requirements of the second definition of a q-dimensional face of a polyhedral set.

2. Now let G be a q-dimensional face of M in the sense of the second definition.

Let the system of restraints defining the polyhedral set G have the form

$$(D_i, X)\leq d_i, \quad i\notin E_G,$$
$$(D_i, X)=d_i, \quad i\in E_G.$$

Consider a point $X \in O$, which can be written in the form

$$X = \alpha X' + (1-\alpha) X'', \quad 0 < \alpha < 1, \quad X', X'' \in M.$$

Obviously, for any l

$$(D_l, X') \leqslant d_l, \tag{1.13}$$

$$(D_l, X'') \leqslant d_l. \tag{1.14}$$

Let $l \in E_O$. Then

$$(D_l, X) = \alpha (D_l, X') + (1-\alpha)(D_l, X'') = d_l,$$

or

$$(D_l, X') = \frac{1}{\alpha}[d_l - (1-\alpha)(D_l, X'')].$$

Further, applying the l-th inequality of (1.14), we obtain

$$(D_l, X') \geqslant \frac{1}{\alpha}[d_l - (1-\alpha) d_l] = d_l.$$

Comparing this relationship with the l-th inequality of (1.13), we observe that

$$(D_l, X') = d_l.$$

Analogously, we obtain

$$(D_l, X'') = d_l.$$

Thus, X', $X'' \in O$. Hence O is a q-dimensional subset of the set M satisfying (b), i. e. , O is a q-dimensional face of M in the sense of the first definition. This completes the proof.

1-4. As a direct corollary of Theorem 1.3 we have the following important characteristic property of the vertices of an arbitrary polyhedral set.

Theorem 1.4. *A point $X \in M$ is a vertex of a polyhedral set if and only if among conditions (1.1), (1.2) there are n linearly independent constraints which reduce to equalities at this point.*

To prove this theorem it is sufficient to recall that a vertex of a polyhedral set is actually equivalent to a zero-dimensional face, and then to apply the second definition of a face of a polyhedral set.

From Theorem 1.4 it follows, in particular, that any polyhedral set has at most a finite number of vertices.

Indeed, a characteristic set of n linearly independent restraints from (1.1), (1.2) corresponds to each vertex of a polyhedral set M; different sets correspond to different vertices. Hence, the number of vertices M is at most C^n_{s+l}.

We shall now discuss the properties of edges (the one-dimensional faces) of a polyhedral set.

Let Γ be some edge of a polyhedral set M. Write out all the rigid constraints imposed on Γ. By assumption, there are exactly $n-1$ linearly independent restraints among them. Hence, the edge Γ is contained in the line of intersection of $n-1$ linearly independent hyperplanes of the form

$$(D_l, X) = d_l, \quad l \in E_\Gamma. \tag{1.15}$$

Let $X_0 \in M$ be on the line (1.15), and let the nonzero vector e point along this line (its components satisfy the homogeneous system corresponding to (1.15)). In this case the equation of the line in question can be written in the form

$$X = X_0 + \lambda e, \quad -\infty < \lambda < \infty. \tag{1.15'}$$

Observe that this representation actually describes the general solution of the system (1.15).

Since all the linearly independent rigid constraints of the edge Γ can be obtained from the system (1.15), X_0 can be considered to reduce all the remaining restraints in (1.1), (1.2) to strict inequalities, i. e.,

$$(D_i, X_0) < d_i, \quad i \notin E_\Gamma \tag{1.16}$$

(those rigid constraints which are linear combinations of equations (1.15) can be neglected).

Since the edge Γ is the intersection of the line (1.15') with the polyhedral set M, it is a set of points X of the form (1.15'), where λ is determined from

$$(D_i, X_0) + \lambda (D_i, e) \leqslant d_i, \quad i \notin E_\Gamma. \tag{1.17}$$

Analyzing the inequalities (1.17), four cases arise:

1. $(D_i, e) = 0$ for all $i \notin E_\Gamma$.
2. $(D_i, e) \geqslant 0$ for all $i \notin E_\Gamma$, and $(D_{i_0}, e) > 0$ for some i_0.
3. $(D_i, e) \leqslant 0$ for all $i \notin E_\Gamma$, and $(D_{i_0}, e) < 0$ for some i_0.
4. (D_i, e) can be either positive or negative.

We shall discuss each of these cases separately.

1. In this case the inequalities (1.17) impose no constraints on the parameter λ. Hence, λ may take on any value between $-\infty$ and ∞, i. e., the edge Γ coincides with the line (1.15'). Observe that in case 1 all the vectors D_i for $i \notin E_\Gamma$ are linear combinations of the vectors D_i for $i \in E_\Gamma$ (see Appendix, Theorem 2.6), since, by assumption, the vectors D_i, $i \notin E_\Gamma$ are orthogonal to the nonzero vector e which is a solution of rank $n-1$ of the system

$$(D_i, e) = 0, \quad i \in E_\Gamma,$$

In this case the rank of the system of restraints of the polyhedral set thus equals $n-1$. In the following we shall see that this condition is not only necessary but also sufficient for the existence of a line edge.

2. If $\lambda \leqslant 0$, then

$$(D_i, X_0) + \lambda (D_i, e) \leqslant (D_i, X_0) < d_i,$$

i. e., the parameter λ is not bounded from below.

The upper bound of λ is, obviously, given by

$$\lambda_0 = \min_{(D_i, e) > 0} \frac{d_i - (D_i, X_0)}{(D_i, e)} > 0, \tag{1.18}$$

where the minimum is taken over those $i \notin E_\Gamma$ for which $(D_i, e) > 0$ (by assumption, the set of these subscripts i is nonempty).

The conditions constraining the range of λ in this case are of the form

$$-\infty < \lambda \leqslant \lambda_0 < \infty, \tag{1.19}$$

i. e., the edge Γ is a half-line (ray) of the form (1.15') with λ obeying the constraints expressed by (1.19).

Consider the end point $\overline{X} = X_0 + \lambda_0 e$ of the ray Γ. It follows from the definition of λ_0 that for some $i' \notin E_\Gamma$

$$(D_{i'}, \overline{X}) = d_{i'},$$

and

$$(D_{i'}, e) > 0.$$

It follows from the last inequality that the vector $D_{i'}$ is not a linear combination of D_i for $i \in E_\Gamma$. The point $\overline{X}$ thus reduces to equalities n linearly

independent restraints of the polyhedral set M (with $l=i'$ and $l \in E_\Gamma$), i. e., $\overline{X}$ is a vertex of M. The equation defining the edge Γ can obviously be written here as

$$X=\overline{X}+\lambda e, \quad \lambda \leqslant 0,$$

where $\overline{X}$ is a vertex of M. If we take $e'=-e$, the equation of the edge Γ can be rewritten in an equivalent form

$$X=\overline{X}+\lambda' e', \quad \lambda' \geqslant 0.$$

The nonzero vector e' pointing along Γ is generally called the direction vector of the edge Γ.

Thus in case 2 the edge Γ is a ray issuing from some vertex of the polyhedral set M.

3. This case is analyzed precisely as the previous one.

The edge Γ is a ray with the equation

$$X=\overline{X}+\lambda e, \quad \lambda \geqslant 0,$$

where $\overline{X}=X_0+\lambda_* e$ is a vertex of M, and e is the direction vector of the edge Γ. Here

$$\lambda_* = \max_{(D_l, e)<0} \frac{d_l-(D_l, X_0)}{(D_l, e)}.$$

4. In this case the parameter λ for which $X_0+\lambda e \in M$ is bounded from above and from below is:

$$\lambda'' \leqslant \lambda \leqslant \lambda'.$$

Here

$$\lambda' = \min_{(D_l, e)>0} \frac{d_l-(D_l, X_0)}{(D_l, e)} > 0,$$

$$\lambda'' = \max_{(D_l, e)<0} \frac{d_l-(D_l, X_0)}{(D_l, e)} < 0.$$

The edge Γ is thus a segment whose end points are

$$\overline{X}=X_0+\lambda' e, \quad \overline{\overline{X}}=X_0+\lambda'' e,$$

and its equation is, consequently, given by

$$X=\lambda\overline{X}+(1-\lambda)\overline{\overline{X}}, \quad 0 \leqslant \lambda \leqslant 1.$$

Reasoning as in case 2, we may easily show that $\overline{X}, \overline{\overline{X}}$ are vertices of the polyhedral set M.

An edge of a polyhedral set which is a line or a half-line is called an unbounded edge. An edge that is a segment will be called a bounded edge.

Unbounded edges may, obviously, occur in unbounded polyhedral sets. All the edges of any polyhedron are segments with the polyhedron vertices as the end points. Thus, using the various properties of edges of polyhedral sets, we formulate the following proposition:

Theorem 1.5. A subset Γ of a polyhedral set M is an edge of M if and only if

(a) Γ is contained in the line of intersection of $n-1$ linearly independent hyperplanes of the form (1.15);

(b) Γ is a line, a ray, or a segment;

(c) the end points of Γ, if any, are vertices of the polyhedral set M.

Proof. The necessary conditions of the theorem have already been established. It remains to prove that these conditions are sufficient. Let Γ satisfy the conditions of the theorem. Consider a polyhedral set M_1 whose restraints are obtained from (1.1), (1.2), where the inequalities for $i \in E_\Gamma$ $(1 \leqslant i \leqslant s)$ are replaced by equalities.

We shall show that $\Gamma = M_1$. The polyhedral set M_1 is the intersection of M with the line (1.15). Obviously, $\Gamma \subset M_1$. We shall verify that $\Gamma = M_1$. If this were not so there would exist a point $X' \in M_1$ outside Γ. Since X' belongs to the line (1.15), the set Γ is either a ray or a segment. Let X'' be the vertex of Γ closest to X'. Then, for sufficiently small $\varepsilon > 0$, the point

$$\overline{X} = X'' + \varepsilon (X'' - X') \in \Gamma.$$

Hence

$$X'' = \frac{\varepsilon}{1+\varepsilon} X' + \frac{1}{1+\varepsilon} \overline{X},$$

and X', $\overline{X} \in M$. Thus, X'' is not a vertex of M, which contradicts (c) of the theorem.

Thus, $\Gamma = M_1$ and, by assumption, $\Gamma = M_1$ is one-dimensional. According to the second definition of an edge of a polyhedral set, Γ is an edge of M. This completes the proof.

§ 2. Representation theorem for a convex polyhedral set

2-1. We have shown that any point of a bounded closed convex set is a convex linear combination of extremum points of this set (the representation theorem). Applying this result to a bounded polyhedral set (a convex polyhedron) and noting that the number of its extremum points, i. e., vertices, is finite, we arrive at the following proposition.

Theorem 2.1. An arbitrary convex polyhedron defined by restraints (1.1), (1.2) is a set of all points X of the form

$$X = \sum_{i=1}^{N} \alpha_i X_i, \tag{2.1}$$

where X_i, $i = 1, 2, \ldots, N$ are the vertices of the polyhedron,

$$\sum_{i=1}^{N} \alpha_i = 1, \quad \alpha_i \geqslant 0.$$

The representation theorem thus specifies entirely the structure of a convex polyhedron. However, the proposition of the theorem is a priori not true for any unbounded set. Indeed, let M be an unbounded polyhedral set. Since M has a finite number of vertices, the set of points which can be written in the form (2.1) is bounded. Hence, it cannot coincide with M. Equation (2.1) is thus unsuitable as a representation of unbounded polyhedral sets. In this section we shall establish a representation theorem for an arbitrary convex polyhedral set.

2-2. First assume that the rank of the system (1.1), (1.2) defining the polyhedral set M is n (equal to the number of components of the vector X). This case is particularly important, since, as we shall show in § 4, any linear-programming problem can be reduced to an equivalent problem whose polyhedral restraint set has this property.

Theorem 2.2. Let the rank of the consistent system (1.1), (1.2) be n. In this case the convex polyhedral set M is a set of all points X of the form

$$X=\sum_{i=1}^{N_1}\alpha_i X_i+\sum_{i=1}^{N_2}\beta_i R_i. \tag{2.2}$$

Here X_i, $i=1,2,\ldots,N_1$ are the vertices of M; R_i, $i=1,2,\ldots,N_2$ are the direction vectors of the unbounded edges of M; all the α_i, β_i are nonnegative; $\sum_{i=1}^{N_1}\alpha_i=1$.

Proof. We shall prove this theorem in two stages. In the first stage we shall establish that any $X\in M$ may be written in the form (2.2). This part of the proof is carried out by induction over the dimensionality ϱ of the polyhedral set M. In the second stage we shall show that any point X which can be written in the form (2.2) belongs to M.

Stage 1. 1. Assume that $\varrho=1$. The polyhedral set M is actually its own edge of maximum dimensionality. Hence, M, being a one-dimensional edge, is either a line, or a ray, or a segment (Theorem 1.5). The first possibility must be neglected since assuming the opposite would imply that the rank of the system defining M is $n-1$. Thus, M is either a ray or a segment. If the former is true, the polyhedral set M can be written in the form

$$X=X_1+\beta_1 R_1, \quad \beta_1\geqslant 0,$$

where X_1 is the end point of the ray M, and R_1 is the direction vector of this ray. If the latter is true, M has the form

$$X=\alpha_1 X_1+\alpha_2 X_2, \quad \alpha_1\geqslant 0, \quad \alpha_2\geqslant 0, \quad \alpha_1+\alpha_2=1,$$

where X_1, X_2 are the end points of segment M. This completes the proof of the theorem for $\varrho=1$

2. Assuming now that the first part of Theorem 2.2 (which asserts that any $X\in M$ can be written in the form (2.2)) is valid for $\varrho\leqslant k-1$, we shall prove its validity for $\varrho=k$. Since the theorem has been proved for $\varrho=1$, we shall assume $k\geqslant 2$. Without loss of generality we may assume that all inequalities (1.1) impose nonrigid constraints on M, and that the vectors D_i, $i=s+1,\ldots,s+t$ are linearly independent. Then, according to Theorem 1.2, $k+t=n$.

By assumption, there are n linearly independent vectors among D_i, $i=1,\ldots,s+t$. Hence, there exist k vectors satisfying the nonrigid constraints of M which, together with the vectors $D_{s+1},\ldots,D_{s+t}$ satisfying the rigid constraints of M, constitute a linearly independent system. Since $k\geqslant 2$, there exist two vectors D_{i_1} and D_{i_2} $(1\leqslant i_1, i_2\leqslant s)$ so that the system D_{i_1}, D_{i_2}, $D_{s+1}, \ldots, D_{s+t}$ is linearly independent.

3. Consider a point $X_0\in M$. First assume that this point reduces all the nonrigid constraints imposed on M to strict inequalities, i. e.,

$$(D_i, X_0)<d_i, \qquad i=1,2,\ldots,s. \tag{2.3}$$

Let $Y=(y_1, y_2,\ldots, y_n)$ be a solution of the linear equations

$$\left.\begin{array}{ll}(D_{i_\lambda}, Y)=(-1)^\lambda, & \lambda=1,2,\\ (D_i, Y)=0, & i=s+1,\ldots, s+t.\end{array}\right\} \tag{2.4}$$

Observe that solvability of (2.4) follows from the linear independence of the vectors D_{i_1}, D_{i_2}, $D_{s+1},\ldots, D_{s+t}$.

Let

$$X(\theta)=X_0+\theta Y.$$

By definition, the vector $X(\theta)$ satisfies (1.2) for any θ. We shall show that this vector satisfies (1.1) only when

$$\theta'\leqslant\theta\leqslant\theta'', \tag{2.5}$$

where $-\infty<\theta'<0$, $0<\theta''<\infty$. Indeed,

$$(D_i,\ X(\theta))=(D_i,\ X_0)+\theta(D_i,\ Y).$$

The inequality $(D_i,\ X(\theta))\leqslant d_i$ is therefore satisfied only when

$$\theta=\begin{cases}\leqslant\dfrac{d_i-(D_i,\ X_0)}{(D_i,\ Y)}, & \text{if } (D_i, Y)>0,\\ \geqslant\dfrac{d_i-(D_i,\ X_0)}{(D_i,\ Y)}, & \text{if } (D_i, Y)<0.\end{cases}$$

Hence, $X(\theta)$ satisfies all the conditions (1.1) if and only if (2.5) holds for

$$\theta'=\max_{(D_i,\ Y)<0}\frac{d_i-(D_i,\ X_0)}{(D_i,\ Y)},$$

$$\theta''=\min_{(D_i,\ Y)>0}\frac{d_i-(D_i,\ X_0)}{(D_i,\ Y)}$$

$$(i=1,2,\ldots,\ s).$$

According to the definition of the vector Y,

$$(D_{i_1},\ Y)<0,\quad (D_{i_2},\ Y)>0$$

Therefore $-\infty<\theta',\ \theta''<\infty$.

It follows from (2.3) that $\theta'<0$, $\theta''>0$. Consider the points $X'=X(\theta')$, $X''=X(\theta'')$. Obviously,

$$X_0=\delta_1X'+\delta_2X'', \tag{2.6}$$

where $\delta_1=\frac{\theta''}{\theta''-\theta'}$, $\delta_2=\frac{\theta'}{\theta'-\theta''}$. The expressions for δ_1 and δ_2 show that $\delta_1\geqslant0$, $\delta_2\geqslant0$, $\delta_1+\delta_2=1$, i. e., the point X_0 is a convex linear combination of the points X' and X''.

Depending on the choice of the numbers θ' and θ'', there exist constraints of M with subscripts i' and i'' so that

$$(D_{i'},\ X')=d_{i'},\quad (D_{i''},\ X'')=d_{i''}.$$

Hence, it follows that X' (X'') belongs to the face M' (M'') of the polyhedral set M whose restraint system is obtained from (1.1), (1.2) when the inequalities i' (i'') are replaced by equalities. Since $(D_{i'},\ Y)\neq0$, $(D_{i''},\ Y)\neq0$, the systems of vectors

$$D_{i'},\ D_{s+1},\ \ldots,\ D_{s+t},$$
$$D_{i''},\ D_{s+1},\ \ldots,\ D_{s+t}$$

are linearly independent. The dimensionalities of the faces M' and M'' are therefore not greater than

$$n-(t+1)=k-1.$$

Applying the assumption of induction, we see that the points $X'\in M'$ and $X''\in M''$ have the form

$$X'=\sum_{i=1}^{N_1'}\alpha_i'X_i'+\sum_{i=1}^{N_2'}\beta_i'R_i', \tag{2.7}$$

$$X'' = \sum_{l=1}^{N_1''} \alpha_l'' X_l'' + \sum_{l=1}^{N_2''} \beta_l'' R_l''. \tag{2.8}$$

Here $X_l'(X_l'')$ are vertices of the polyhedral set $M'(M'')$; $R_l'(R_l'')$ are the direction vectors of the unbounded edges $M'(M'')_{N_1'}$. All the $\alpha_l', \alpha_l'', \beta_l', \beta_l''$ are nonnegative;

$$\sum_{l=1}^{N_1'} \alpha_l' = 1, \ \sum_{l=1}^{N_l''} \alpha_l'' = 1.$$

4. We shall now show that the points X_l', X_l'' are vertices of the polyhedral set M, and R_l' and R_l'' are the direction vectors of the unbounded edges of this set. We observe that if M_2 is a face of the polyhedral set M_1, which in turn is a face of the polyhedral set M, then M_2 is a face of M. Indeed, let

$$X = \gamma_1 X_1 + \gamma_2 X_2, \tag{2.9}$$

where $X \in M_2$; $X_1, X_2 \in M$; $\gamma_1 + \gamma_2 = 1$; $\gamma_1, \ \gamma_2 \geqslant 0$. Since $M_2 \subset M_1$, we have $X \in M_1$. Hence, according to the first definition of a face, we observe that $X_1, X_2 \in M_1$. Applying this definition to the face M_2 of the polyhedral set M_1, we obtain the desired inclusion, $X_1, X_2 \in M_2$.

Thus, a point $X \in M_2$ has the form (2.9) only if the points X_1, X_2 entering the representation also belong to M_2.

Applying again the first definition of a face, we conclude that M_2 is a face of M. Hence, in particular, follows the previous proposition regarding the points X_l', X_l'' and the vectors R_l', R_l''.

Substituting in (2.6) X' and X'' as expressed by (2.7) and (2.8), respectively, we obtain X_0 as a sum of a convex linear combination of the points X_l' and X_l'' and a nonnegative linear combination of the vectors R_l' and R_l''. Since X_l', X_l'' are vertices of M, and R_l', R_l'' are direction vectors of the unbounded edges of M, we conclude that any point X_0 satisfying the inequalities (2.3), can be written in the form (2.2).

5. Suppose that $X_0 \in M$ does not satisfy (2.3), i. e., there exists an r-th nonrigid constraint of $M (1 \leqslant r \leqslant s)$ so that

$$(D_r, \ X) = d_r. \tag{2.10}$$

In this case X_0 belongs to the polyhedral set M_0 defined by (1.1) with $l \neq r$, (1.2), and the equality (2.10).

Observe that the system of vectors

$$D_r, \ D_{s+1}, \ D_{s+2}, \ \ldots, \ D_{s+t} \tag{2.11}$$

is linearly independent. Since by assumption the r-th inequality of the system (1.1) is a nonrigid constraint of M, there exists a point $X' \in M$ so that

$$(D_r, \ X') = d < d_r.$$

Let

$$Y = X_0 - X'.$$

The vector Y satisfies the equations

$$\left.\begin{aligned} &(D_l, \ Y) = 0, \qquad l = s+1, \ \ldots, \ s+t, \\ &(D_r, \ Y) = d_r - d > 0. \end{aligned}\right\} \tag{2.12}$$

Relationships (2.12) indicate that D_r cannot be represented as a linear combination of the vectors $D_{s+1}, \ \ldots, \ D_{s+t}$. Further, noting that the vectors satisfying the rigid constraints of M are linearly independent, we confirm the

linear independence of the vectors (2.11). Thus, the rank of the system of rigid constraints imposed on the polyhedral set M_0 is not less than $t+1$ and, consequently, the dimensionality of this set is at most

$$n-(t+1)=k-1.$$

As in point 3 of the proof, applying the assumption of induction we verify that X can be represented in the form (2.2).

We have thus shown by induction that any point of a polyhedral set M, whose dimensionality ϱ is equal to $k\geqslant 2$, can be written in the form (2.2). Since the theorem applies for $\varrho=1$, this completes the proof of the first part of the theorem.

Stage 2. 6. It now remains to show that any point, which can be written in the form of (2.2), belongs to the polyhedral set M.

We first note that $N_1\geqslant 1$. Otherwise, (2.2) would contain only the vectors R_i. The set of all these vectors is nonempty, since, by assumption, there exists a point $X\in M$ which, according to the first part of the theorem, can be written in the form (2.2). Hence, M has unbounded edges each of which, in this case, is a ray (by assumption, the rank of the system (1.1), (1.2) is n). The end point of any such ray is a vertex of M. Therefore, contrary to our assumption, M has vertices. Thus, a polyhedral set M always has extremum points.

7. Let R be the direction vector of any unbounded edge of the polyhedral set M.

In our analysis of unbounded edges of polyhedral sets (§ 1) we have shown that

$$(D_i,\ R)\leqslant 0,\qquad i=1,\ 2,\ \ldots,\ s, \tag{2.13}$$
$$(D_i,\ R)=0,\qquad i=s+1,\ \ldots,\ s+t. \tag{2.14}$$

Consider now a point X_0 of the form

$$X_0=\sum_{i=1}^{N_1}\alpha_i X_i+\sum_{i=1}^{N_2}\beta_i R_i,$$

where all α_i, β_i are nonnegative and $\sum\limits_{i=1}^{N_1}\alpha_i=1$. We shall show that $X_0\in M$. In fact,

$$(D_i,\ X_0)=\sum_{k=1}^{N_1}\alpha_k(D_i,\ X_k)+\sum_{k=1}^{N_2}\beta_k(D_i,\ R_k).$$

Applying the inequalities (1.1), which the points X_k satisfy, and the inequalities (2.13), which hold for each vector R_k, we obtain for $1\leqslant i\leqslant s$

$$(D_i,\ X_0)\leqslant d_i\sum_{i=1}^{N_1}\alpha_i=d_i.$$

With $s+1\leqslant i\leqslant s+t$, we obtain from (1.2) and (2.14)

$$(D_i,\ X_0)=d_i\sum_{i=1}^{N_1}\alpha_i=d_i.$$

Point X_0 thus satisfies all the conditions (1.1), (1.2), i.e., it belongs to the polyhedral set M. This completes the proof of the second part of the theorem.

Theorem 2.2 contains, as a particular case, the result of Theorem 2.1. Indeed, if M is a convex polyhedron, M has no unbounded edges, i.e., $N_2=0$.

In this case the representation (2.2) reduces to the representation (2.1) of Theorem 2.1.

In our proof of Theorem 2. 2 (point 6) we established an important proposition, which may be summed up in the following theorem.

Theorem 2.3. If the rank of the restraint system (1.1), (1.2) defining a polyhedral set M is n, M has at least one vertex.

According to Theorem 1. 4 to each vertex of a polyhedral set M correspond n linearly independent restraints from (1.1), (1.2), which are reduced to equalities at this point. Therefore, the existence of at least one vertex of M shows that the rank of the system (1.1), (1.2) is n. Combining this result with Theorem 2.3, we arrive at the following result:

Theorem 2.4. For a nonempty polyhedral set M to have at least one vertex, it is necessary and sufficient that the rank of the restraint system (1.1), (1.2) be n.

2-3. Theorem 2.2 establishes the structure of a polyhedral set M for the case when the rank of the system (1.1), (1.2) is exactly n (the dimensionality of the space of points X). Let us now investigate the general case, when the rank of the system (1.1), (1.2) is $r \leqslant n$.

Let the rank of the system of vectors D_i, $i=1, 2, \ldots, s+t$, be $r<n$. This indicates that the matrix $\|d_{ij}\|_{s+t, n}$ has r linearly independent columns, the remaining $n-r$ columns being linear combinations of the other columns.

Let the j-th column of the matrix $\|d_{ij}\|_{s+t, n}$ be denoted by

$$D^{(j)}=(d_{1j}, d_{2j}, \ldots, d_{s+t, j})^T, \qquad j=1, 2, \ldots, n. \tag{2.15}$$

The restraint system (1.1), (1.2) can now be written in the form

$$\sum_{j=1}^{n} x_j D^{(j)} \leqq D. \tag{2.16}$$

Here $D=(d_1, d_2, \ldots, d_{s+t})^T$; the sign $\leqq$ indicates that the first s components of the vector relationship (2.16) are inequalities, and the subsequent t components are equalities.

By assumption, the maximum linearly independent subsystem of the system $D^{(j)}$, $j=1, 2, \ldots, n$ comprises r vectors. Without loss of generality we may take these to be the first r vectors of the system (2.15)

$$D^{(1)}, D^{(2)}, \ldots, D^{(r)}. \tag{2.17}$$

We now express the vectors $D^{(j)}$ with $j>r$ as linear combinations of the vectors (2.17)

$$D^{(j)}=\sum_{i=1}^{r} \delta_{ij} D^{(i)}. \tag{2.18}$$

Applying the equality (2.18), we reduce the restraint system (2.16) to the equivalent form

$$\sum_{i=1}^{r} \left(x_i+\sum_{j=r+1}^{n} x_j \delta_{ij}\right) D^{(i)} \leqq D. \tag{2.19}$$

Consider the polyhedral set M_0 whose restraint system comprises (2.16), or (2.19), and the equations

$$x_{r+1}=x_{r+2}=\ldots=x_n=0. \tag{2.20}$$

We easily observe that the rank of the restraint system of the polyhedral set M_0 equals exactly n.

We now establish a relationship between the polyhedral sets M and M_0, which will enable us to apply the previous results to the analysis of the structure of M.

Let Π_{n-r} be an $(n-r)$-dimensional subspace generated by the intersection of r, $(n-1)$-dimensional subspaces (hyperplanes) defined by linearly independent homogeneous equations

$$L_i(X)=x_i+\sum_{j=r+1}^{n}\delta_{ij}x_j=0, \qquad i=1,\ 2,\ \ldots,\ r. \tag{2.21}$$

Lemma 2.1. A polyhedral set M is the sum of the polyhedral set M_0 and the subspace Π_{n-r}:*

$$M=M_0+\Pi_{n-r}.$$

Proof. 1. Let $X'\in M_0$, $X''\in\Pi_{n-r}$. We shall show that in this case

$$X=X'+X''\in M.$$

Representing the restraints defining M in the form (2.19), we obtain

$$\sum_{j=1}^{n}x_jD^{(j)}=\sum_{i=1}^{r}(x_i+\sum_{j=r+1}^{n}\delta_{ij}x_j)D^{(i)}=\sum_{i=1}^{r}L_i(X)D^{(i)}=\sum_{i=1}^{r}L_i(X')D^{(i)}+\sum_{i=1}^{r}L_i(X'')D^{(i)}.$$

By assumption,

$$\sum_{i=1}^{r}L_i(X')D^{(i)}\leqq D \qquad (X'\in M_0);$$

$$L_1(X'')=\ldots=L_r(X'')=0 \ (X''\in\Pi_{n-r}),$$

whence

$$\sum_{j=1}^{n}x_jD^{(j)}\leqq D,$$

i. e., $X\in M$.

2. Let now X be a point in M. Let

$$x_j'=\begin{cases} L_j(X) & \text{for } j=1,\ 2,\ \ldots,\ r, \\ 0 & \text{for } j=r+1,\ \ldots,\ n, \end{cases}$$

$$x_j''=x_j-x_j', \quad i=1,\ 2,\ \ldots,\ n.$$

Applying (2.19), we readily verify that

$$X'=(x_1',\ x_2',\ \ldots,\ x_n')\in M_0.$$

From the definition of the points X' and $X''=(x_1'',\ x_2'',\ \ldots,\ x_n'')$ we have

$$L_i(X'')=L_i(X)-L_i(X')=L_i(X)-x_i'=$$
$$=L_i(X)-L_i(X)=0,\ i=1,\ 2,\ \ldots,\ r.$$

Hence

$$X''\in\Pi_{n-r}.$$

Any point $X\in M$ can, thus, be represented as the sum of points $X'\in M_0$ and $X''\in\Pi_{n-r}$. This completes the proof.

In particular, it follows from Lemma 2.1 that the equality $r=n$ is a necessary condition for the boundedness of a polyhedral set M.

Let us now establish a correspondence between the faces of the polyhedral sets M and M_0.

We shall say that the faces $\Gamma\in M$ and $\Gamma_0\in M_0$ are **corresponding faces** if the restraints defining Γ_0 differ from the restraint system of the face Γ only by the additional equalities (2.20).

* By definition, a set A is the sum of sets B and C $(A=B+C)$, if $X\in A$ can be written as $X=Y+Z$, where $Y\in B$, $Z\in C$.

According to the Lemma 2.1, the face Γ is the sum of the face Γ_0 and the $(n-r)$-dimensional space Π_{n-r}. It can easily be verified (see Exercise 4) that the dimensionalities $\varrho(\Gamma)$ and $\varrho(\Gamma_0)$ of the faces Γ and Γ_0, respectively, are related by the expression

$$\varrho(\Gamma)=\varrho(\Gamma_0)+n-r, \tag{2.22}$$

where r is the rank of the restraint system defining the set M. Applying Theorem 2.4 to the polyhedral set M_0 and using equality (2.22), we arrive at the following proposition.

Theorem 2.5. (Generalization of Theorem 2.4.) The minimum dimensionality of the faces of a polyhedral set M is $n-r$, where r is the rank of the restraint system (1.1), (1.2).

We note that according to Lemma 2.1 an arbitrary face of minimum dimensionality in M coincides with the intersection of the hyperplanes

$$x_i+\sum_{j=r+1}^{n}\delta_{ij}x_j=\bar{x}_i+\sum_{j=r+1}^{n}\delta_{ij}\bar{x}_j,\quad i=1,\ 2,\ \ldots,\ r, \tag{2.23}$$

where $\bar{X}=(\bar{x}_1,\ \bar{x}_2,\ \ldots,\ \bar{x}_n)$ is some vertex of the set M_0.

Lemma 2.1 makes it possible to extend the principal theorem of this section, Theorem 2.2, to the case of an arbitrary polyhedral set.

Theorem 2.6. (Representation theorem of polyhedral sets.) An arbitrary polyhedral set M is a set of all points X of the form

$$X=\sum_{i=1}^{N_1}\alpha_i X_i+\sum_{i=1}^{N_2}\beta_i R_i+\sum_{i=1}^{n-r}\gamma_i R_i', \tag{2.24}$$

where $\sum_{i=1}^{N_1}\alpha_i=1$; $\alpha_i\geqslant 0$, $\beta_i\geqslant 0$; X_i are the vertices of M_0; R_i are the direction vectors of the unbounded edges of M_0; R_i', $i=1,\ 2,\ \ldots,\ n-r$ are any complete system of linearly independent vectors of the subspace Π_{n-r}; r is the rank of the restraint system defining M.

Theorem 2.6 is an obvious consequence of Theorem 2.2 and Lemma 2.1.

2-4. To illustrate the results of this section, we consider two numerical examples.

Example 1. Consider a polyhedral set M_1 defined by the restraints

$$\left.\begin{aligned} -x_1-x_2&\leqslant 0,\\ -x_1+x_2&\leqslant 0,\\ -x_2&\leqslant 2,\\ -x_1+2x_2&\leqslant 2.\end{aligned}\right\} \tag{2.25}$$

The image of M_1 in the x_1Ox_2-plane is shown in Figure 2.1. We observe from Figure 2.1 that the faces of minimum dimensionality of M_1 are the points $X_1=(2,-2)$, $X_2=(2,\ 2)$, and $X_3=(0,\ \ 0)$. This conculsion is in agreement with Theorem 2.4, since the rank of the system (2.25) is $n=2$ (the vectors $D_1=(-1,-1)$, $D_2=(-1,\ 1)$ are linearly independent). The polyhedral set M_1 is not bounded and, consequently, it should have unbounded edges. We observe from Figure 2.1 that this set has two unbounded edges issuing from points X_1 and X_2 and having $R_1=(1,\ 0)$ and $R_2=(2,\ 1)$ as their direction vectors.

According to Theorem 2.2, the polyhedral set M_1 can be represented as the sum of an arbitrary convex combination of the points X_1, X_2, X_3 and an arbitrary nonnegative combination of the vectors R_1 and R_2, i. e., M_1 is a set of all the points

$$X=\alpha_1(2,\ -2)+\alpha_2(2,\ 2)+\alpha_3(0,\ 0)+\beta_1(1,\ 0)+\beta_2(2,\ 1),$$

where $\alpha_i\geqslant 0$, $\beta_i\geqslant 0$, $\alpha_1+\alpha_2+\alpha_3=1$, or

$$X=(2\alpha_1+2\alpha_2+\beta_1+2\beta_2,\ -2\alpha_1+2\alpha_2+\beta_2),$$

where $\alpha_1\geqslant 0$, $\alpha_2\geqslant 0$, $\beta_1\geqslant 0$, $\beta_2\geqslant 0$, $\alpha_1+\alpha_2\leqslant 1$.

Example 2. Consider a polyhedral set M defined by the restraints

$$\begin{aligned} -x_1-x_2-2x_3 &\leqslant 0, \\ -x_1+x_2 \qquad &\leqslant 0, \\ -x_2 \qquad -x_3 &\leqslant 2, \\ -x_1+2x_2+x_3 &\leqslant 2. \end{aligned} \tag{2.26}$$

Here $n=3$; it is, however, easily verified that the rank of (2.26) is 2. Let

$$\begin{aligned} D^{(1)} &= (-1,\ -1,\ 0,\ -1)^T, \\ D^{(2)} &= (-1,\ 1,\ -1,\ 2)^T, \\ D^{(3)} &= (-2,\ 0,\ -1,\ 1)^T. \end{aligned}$$

The vectors $D^{(1)}$ and $D^{(2)}$ are obviously linearly independent, whereas $D^{(3)}=D^{(1)}+D^{(2)}$.

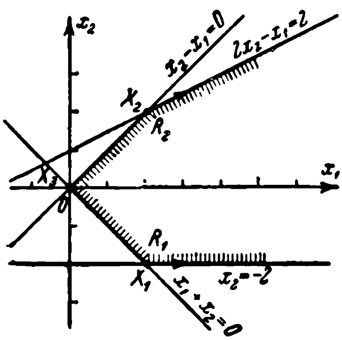


FIGURE 2.1

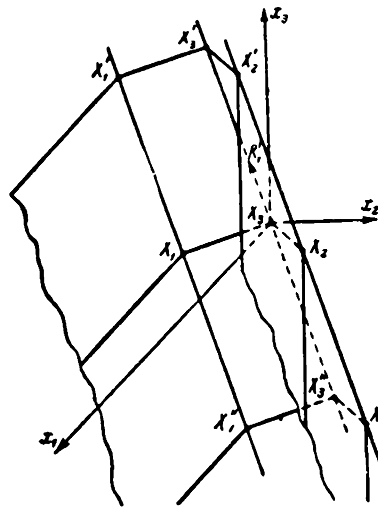


FIGURE 2.2

According to the general rule, the polyhedral set M_0 is defined by the restraints (2.26) and the additional equality

$$x_3=0.$$

We observe that M_0 coincides with the polyhedral set M_1 discussed in the preceding example. Since $n-r=1$ here, the subspace $\Pi_{n-r}=\Pi_1$ is a line. Transforming (2.26) to the form

$$\begin{aligned} -(x_1+x_3)-(x_2+x_3) &\leqslant 0, \\ -(x_1+x_3)+(x_2+x_3) &\leqslant 0, \\ -(x_2+x_3) &\leqslant 2, \\ -(x_1+x_3)+2(x_2+x_3) &\leqslant 2, \end{aligned}$$

we observe that this line is defined by

$$x_1+x_3=0, \quad x_2+x_3=0. \tag{2.27}$$

According to Lemma 2.1 the polyhedral set M can be considered as consisting of all the lines parallel to (2.27), and passing through points of M_0. The image of M is given in Figure 2.2.

Since $n-r=1$, the minimum dimensionality of the faces of M is 1. The polyhedral set M has three faces of minimum dimensionality: these are $X_1'X_1''$, $X_2'X_2''$, and $X_3'X_3''$. Each of these lines corresponds to a definite vertex of M_0. Applying formula (2.24), we give the general representation of the points $X\in M$.

In this case $M_0=M_1$. Therefore, $X_1=(2,-2,0)$, $X_2=(2,2,0)$, $X_3=(0,0,0)$, $R_1=(1,0,0)$, $R_2=(2,1,0)$. The vector R_1' (the only one, since $n-r=1$) is the direction vector of the line (2.27)

$$R_1'=(-1,\ -1,\ 1).$$

Thus, the polyhedral set M is a set of all the points X of the form

$$X=\alpha_1X_1+\alpha_2X_2+\alpha_3X_3+\beta_1R_1+\beta_2R_2+\gamma_1R_1'$$

or

$$X=(2\alpha_1+2\alpha_2+\beta_1+2\beta_2-\gamma_1,\ -2\alpha_1+2\alpha_2+\beta_2-\gamma_1,\ \gamma_1),$$

where

$$\alpha_1\geqslant 0,\quad \alpha_2\geqslant 0,\quad \beta_1\geqslant 0,\quad \beta_2\geqslant 0,\quad \alpha_1+\alpha_2\leqslant 1.$$

§ 3. Equivalence of the two definitions of a convex polyhedral set

3-1. Theorem 2.2, establishing representation (2.2) for any polyhedral set with $r=n$, can be stated in alternative form:

Any polyhedral set M with $r=n$ can be written as a sum of a convex combination of the vertices of M and a nonnegative linear combination of the direction vectors of the unbounded edges of M.

It is easily observed that any s-dimensional subspace is a nonnegative linear combination of a finite number of points. Indeed, if $X_1, X_2, \ldots, X_s$ is an arbitrary linearly independent system of vectors in this subspace, the desired system may be taken as comprising all the points of the form

$$X_1,\ X_2,\ \ldots,\ X_s;\ -X_1,\ -X_2,\ \ldots,\ -X_s.$$

This system of points consists of $2s$ elements. It can be shown that the smallest system having the same properties contains only $s+1$ points (Exercise 5). We can now advance a new formulation of the general representation theorem for a polyhedral set (Theorem 2.6), analogous to the alternative formulation of Theorem 2.2.

Any polyhedral set can be written as a sum of convex and nonnegative linear combinations of some systems of points. In this section we shall establish the converse proposition; this will introduce a new definition of a polyhedral set which is useful in some theoretical problems of linear programming.

3-2. We consider a certain class of convex polyhedral sets, significant in the following (for the definition of a convex cone in any finite-dimensional space, see Appendix, 3-2).

A polyhedral convex cone is a convex polyhedral set having the properties of a cone. In other words a polyhedral set M defined by the restraints (1.1), (1.2) is a polyhedral cone with apex at X_0 (the adjective "convex" is omitted for the sake of brevity) if from $X\in M$ it follows that

$$X_0+\lambda(X-X_0)\in M$$

for any $\lambda\geqslant 0$.

Let M be a polyhedral cone with apex at X_0 and let $X\in M$. Then, for any i, $1\leqslant i\leqslant s$,

$$(D_i,\ X-X_0)\leqslant 0.$$

Otherwise there exist $\lambda>0$ such that for some i

$$(D_i,\ X_0+\lambda(X-X_0))>d_i.$$

This inequality is not in accordance with the property of the cone M

$$X_0+\lambda(X-X_0)\in M\quad\text{for any}\quad \lambda\geqslant 0.$$

Since $X_0 \in M$,

$$(D_i,\ X - X_0) = 0$$

for $s \leqslant i \leqslant s+t$. Thus, any $X \in M$ satisfies the restraints

$$(D_i,\ X) \leqslant (D_i,\ X_0), \qquad i = 1,\ 2,\ \ldots,\ s, \tag{3.1}$$

$$(D_i,\ X) = (D_i,\ X_0), \qquad i = s+1,\ \ldots,\ s+t. \tag{3.2}$$

An arbitrary point obeying (3.1), (3.2), obviously satisfies (1.1), (1.2) and as such is contained in a polyhedral cone M.

The restraints defining a polyhedral cone may thus be reduced to the form (3.1), (3.2). It can easily be shown that any set defined by the restraints (3.1), (3.2) is a polyhedral cone.

Let K be the polyhedral cone defined by the restraints (3.1), (3.2). Since the apex X_0 of the cone K satisfies the equations

$$(D_i,\ X) = d_i = (D_i,\ X_0), \qquad i = 1,\ 2,\ \ldots,\ s+t,$$

the vector $D = (d_1,\ d_2,\ \ldots,\ d_{s+t})^T$ is a linear combination of the vectors

$$D^{(j)},\ j = 1,\ 2,\ \ldots,\ n.$$

The rank of the system of vectors

$$D,\ D^{(j)},\ j = 1,\ 2,\ \ldots,\ n,$$

thus equals the rank of the subsystem

$$D^{(j)},\ j = 1,\ 2,\ \ldots,\ n.$$

Further, noting that the ranks of the systems

$$D_i,\ i = 1,\ 2,\ \ldots,\ s+t;\quad \tilde{D}_i = (d_{i1},\ \ldots,\ d_{in},\ d_i),$$
$$i = 1,\ 2,\ \ldots,\ s+t,$$

are equal, respectively, to the ranks of the systems

$$D^{(j)},\ j = 1, 2,\ \ldots,\ n;\quad D,\ D^{(j)},\ j = 1,\ 2,\ \ldots,\ n,$$

we conclude that the vector systems

$$D_i,\ i = 1, 2,\ \ldots,\ n \text{ and } \tilde{D}_i,\ i = 1,\ 2,\ \ldots,\ n$$

have the same rank. Therefore, if $D_{i_1}, \ldots, D_{i_r}$ is the largest linearly independent subsystem of the vectors D_i, $i = 1, 2, \ldots, s+t$, the vectors $\tilde{D}_{i_1}, \ldots, \tilde{D}_{i_r}$ constitute the largest linearly independent subsystem of the vector system

$$\tilde{D}_i,\ i = 1,\ 2,\ \ldots,\ s+t.$$

Hence, the point X satisfying the system of equations

$$(D_{i_\alpha},\ X) = d_{i_\alpha}, \quad \alpha = 1,\ 2, \ldots, r, \tag{3.3}$$

where D_{i_α}, $\alpha = 1, 2, \ldots, r$ is the largest linearly independent subsystem of the system of vectors D_i, $i = 1, 2, \ldots, s+t$, reduces all the restraints (3.1), (3.2) to equalities.

We can now easily establish the uniqueness of the minimum face of the polyhedral cone K. If X is a point belonging to some minimum face of the cone K, whose system of defining restraints is of rank r, it should satisfy equations of the form (3.3) (Theorem 2.5) and as such reduce all the restraints (3.1), (3.2) to equalities. Hence, any minimum face of the cone K coincides with the set of solutions of the system $(D_i,\ X) = d_i$, $i = 1, 2, \ldots, s+t$.

Therefore, the polyhedral cone K has but one face of minimum dimensionality. According to Theorem 2.5 the minimum face of the cone K is $n-r$-dimensional, where r is the rank of the restraint system (3.1), (3.2). In particular, for $r=n$, the point X_0, the apex of the cone K, is the only extremum point of K. In this case X_0 is generally termed the p o i n t of the cone K.

Observe that, in general, the apex of a polyhedral cone is not equivalent to a vertex of a polyhedral set. The two concepts coincide only if the apex of the cone is also the point of the cone. Otherwise, any point belonging to the minimum face of the cone can be taken as its apex. Obviously, none of these points satisfies the definition of a vertex of the polyhedral set representing the cone.

This inconsistency in concepts is due to the fact that terminology used today for cones and polyhedral sets developed independently. However, no confusion should arise.

The apex of a cone will be taken as an arbitrary point belonging to the face of minimum dimensionality.

Applying Theorem 2.6, we may assert that an arbitrary convex polyhedral cone K comprises all the points X of the form

$$X=X_0+\sum_{l=1}^{N}\beta_l R_l, \tag{3.4}$$

where $\beta_l \geqslant 0$, $l=1, 2, \ldots, N_1$; X_0 is the apex of K; R_l is a set of vectors.

3-3. Consider the set T comprising the points X written as

$$X=\sum_{l=1}^{N_1}\alpha_l \; X_l+\sum_{l=1}^{N_2}\beta_l R_l, \tag{3.5}$$

where $\alpha_l \geqslant 0$, $\beta_i \geqslant 0$, $\sum_{l=1}^{N_1}\alpha_l=1$. The primary purpose of this section is to prove that T is a convex polyhedral set. To this end we first establish three lemmas.

Lemma 3.1. Let the vectors $A_1, A_2, \ldots, A_k$ *be linearly independent and let*

$$\left|\sum_{l=1}^{k}\alpha_l A_l\right| \leqslant c_1.$$

There exists a number c_2 *independent of the coefficients* α_i *so that*

$$\sum_{l=1}^{k}|\alpha_i|<c_2.$$

Proof. Suppose the contrary is true, i. e., let there exist a sequence of vectors

$$\alpha^{(s)}=(\alpha_1^{(s)}, \; \alpha_2^{(s)}, \; \ldots, \; \alpha_k^{(s)}),$$

so that

$$\left|\sum_{l=1}^{k}\alpha_i^{(s)} A_i\right| \leqslant c_1 \tag{3.6}$$

and, moreover,

$$\lim_{s\to\infty}\sum_{l=1}^{k}\left|\alpha_i^{(s)}\right|=\infty. \tag{3.7}$$

Let

$$\overline{\alpha}^{(s)}=\frac{1}{\sum_{l=1}^{k}|\alpha_i^{(s)}|}\alpha^{(s)}.$$

From (3.6) and (3.7) we have

$$\lim_{s\to\infty}\left|\sum_{i=1}^{k}\bar{\alpha}_i^{(s)}A_i\right|=0. \tag{3.8}$$

From the definition of $\bar{\alpha}^{(s)}$, we have

$$\sum_{i=1}^{k}|\bar{\alpha}_i^{(s)}|=1 \tag{3.9}$$

for any s. Therefore, from the sequence $\{\bar{\alpha}^{(s)}\}$ we may choose a convergent subsequence $\{\bar{\alpha}^{(s_t)}\}_t$

$$\lim_{t\to\infty}\bar{\alpha}_i^{(s_t)}=\bar{\alpha}_i, \qquad i=1,2,\ \ldots,\ k.$$

Hence, applying (3.8), we have

$$\sum_{i=1}^{k}\bar{\alpha}_i A_i=0.$$

Since the vectors $A_1, A_2, \ldots, A_k$ are linearly independent,

$$\bar{\alpha}_1=\bar{\alpha}_2=\ldots=\bar{\alpha}_k=0.$$

However, from (3.9)

$$\sum_{i=1}^{k}|\bar{\alpha}_i|=1.$$

This contradiction proves the lemma.

Lemma 3.2. For any point X, which can be written in the form (3.5), there exist linearly independent vectors $R_{i_1}, R_{i_2}, \ldots, R_{i_l}$ *of the system* R_i, $i=1, 2, \ldots, N_2$, *so that*

$$X=\sum_{i=1}^{N_1}\alpha_i X_i+\sum_{\lambda=1}^{l}\bar{\beta}_\lambda R_{i_\lambda}, \tag{3.10}$$

where $\alpha_i\geqslant 0$, $\bar{\beta}_{i_\lambda}>0$, $\sum_{i=1}^{N_1}\alpha_i=1$.

Proof. Consider the representation (3.5) of the point X with the minimum number of positive coefficients β_i. Let (3.10) be one of these representations. We shall show that in this case the vectors $R_{i_1}, R_{i_2}, \ldots, R_{i_l}$ are linearly independent.

Suppose the vectors are linearly dependent, then

$$\sum_{\lambda=1}^{l}t_\lambda R_{i_\lambda}=0, \tag{3.11}$$

where not all t_λ are zero. Without loss of generality we assume that one of the $t_\lambda>0$ (if this requirement is not satisfied, we may multiply both sides of (3.11) by -1). Multiplying (3.11) by $\theta>0$ and subtracting the result from (3.10), we obtain

$$X=\sum_{i=1}^{N_1}\alpha_i X_i+\sum_{\lambda=1}^{l}(\bar{\beta}_\lambda-\theta t_\lambda)R_{i_\lambda}.$$

If we now take

$$\theta=\min_{t_\lambda>0}\frac{\bar{\beta}_\lambda}{t_\lambda},$$

we obtain a new representation of the point X with the number of positive β_i not exceeding

$$l-1<l.$$

2154

This contradiction shows that the assumption of dependence, equality (3.11), is false; the system of vectors

$$R_{l_1},\ R_{l_2},\ \ldots,\ R_{l_l}$$

is linearly independent. This completes the proof.

Lemma 3.3. The set T *of points* X, *which can be written in the form (3.5), is closed.*

Proof. Consider an arbitrary convergent sequence of points $X^{(t)} \in \mathrm{T}$. To prove the lemma, we must show that

$$\lim_{t\to\infty} X^{(t)} = X^{(0)} \in \mathrm{T}. \tag{3.12}$$

Since $X^{(t)}$ is an element of T for any t,

$$X^{(t)} = \sum_{l=1}^{N_1} \alpha_l^{(t)} X_l + \sum_{\lambda=1}^{l_t} \beta_{\lambda,t} R_{l_{\lambda,t}},$$

where $\alpha_l^{(t)} \geqslant 0$, $\beta_{\lambda,t} > 0$, $\sum_{l=1}^{N_1} \alpha_l^{(t)} = 1$.

According to Lemma 3.2, the system of vectors $R_{l_{\lambda,t}}$, $\lambda = 1, 2, \ldots, l_t$ comprising some of the vectors R_l, $l = 1, 2, \ldots, N_2$ is linearly independent for any t.

The number of points $X^{(t)}$ is infinite and the number of possible systems which can be constructed from the vectors $R_1, R_2, \ldots, R_{N_2}$ is a priori finite. Therefore, there exists an infinite subsequence $\{X^{(t_r)}\}_r$ of the sequence $\{X^{(t)}\}$ so that

$$X^{(t_r)} = \sum_{l=1}^{N_1} \alpha_l^{(r)} X_l + \sum_{\lambda=1}^{l} \beta_\lambda^{(r)} R_{l_\lambda}, \tag{3.13}$$

where $R_{l_1}, R_{l_2}, \ldots, R_{l_l}$ is a linearly independent system of vectors, which is the same for all the points $X^{(t_r)}$, $\alpha_l^{(r)} \geqslant 0$, $\beta_\lambda^{(r)} > 0$, $\sum_{l=1}^{N_1} \alpha_l^{(r)} = 1$.

Since the subsequence $\{X^{(t_r)}\}_r$ converges, there exists a number c_1 so that

$$|X^{(t_r)}| \leqslant c_1$$

for any r. Therefore, applying the principal properties of vector norms (lengths), we obtain for any r

$$\begin{aligned}\left|\sum_{\lambda=1}^{l} \beta_\lambda^{(r)} R_{l_\lambda}\right| &= \left|X^{(t_r)} - \sum_{l=1}^{N_1} \alpha_l^{(r)} X_l\right| \leqslant \left|X^{(t_r)}\right| + \\ &+ \left|\sum_{l=1}^{N_1} \alpha_l^{(r)} X_l\right| \leqslant c_1 + \left(\sum_{l=1}^{N_1} \alpha_l^{(r)}\right) c_2 = c_1 + c_2 = c_3,\end{aligned} \tag{3.14}$$

where

$$c_2 = \max_{1\leqslant l\leqslant N_1} |X_l|.$$

According to Lemma 3.1, we obtain from the inequality (3.14) a constant c_4 independent of r so that

$$\left|\sum_{\lambda=1}^{l} \beta_\lambda^{(r)}\right| \leqslant c_4.$$

Thus, in (3.14), not only the numbers $\alpha_l^{(r)}\,(0 \leqslant \alpha_l^{(r)} \leqslant 1)$ are bounded, but also the coefficients $\beta_\lambda^{(r)}\,(0 \leqslant \beta_\lambda^{(r)} \leqslant c_4)$. This proves the existence of convergent subsequences of the sequences $\{\alpha_l^{(r)}\}_r$, $l = 1, 2, \ldots, N_1$; $\{\beta_\lambda^{(r)}\}_r$, $\lambda = 1, 2, \ldots, l$.

Hence, there exists a subsequence of indices $r_1, r_2, \ldots, r_\gamma, \ldots$ so that

$$\lim_{\gamma\to\infty} \alpha_i^{(r_\gamma)} = \alpha_i^{(0)}, \qquad i=1, 2, \ldots, N_1,$$
$$\lim_{\gamma\to\infty} \beta_\lambda^{(r_\gamma)} = \beta_\lambda^{(0)}, \qquad i=1,2, \ldots, l.$$

Obviously,

$$\alpha_i^{(0)} \geqslant 0, \ i=1, 2, \ldots, N_1; \quad \beta_\lambda^{(0)} \geqslant 0, \quad \lambda=1,2, \ldots, l;$$
$$\sum_{i=1}^{N_1} \alpha_i^{(0)} = 1.$$

Now, if in (3.13) r tends to infinity over the subsequence r_γ, we obtain

$$\sum_{i=1}^{N_1} \alpha_i^{(0)} X_i + \sum_{\lambda=1}^{l} \beta_\lambda^{(0)} R_{i_\lambda} = X^{(0)}.$$

Hence, applying the properties of the coefficients $\alpha_i^{(0)}$, $\beta_\lambda^{(0)}$, we obtain (3.12), which proves the lemma.

3-4. We now formulate and prove the converse representation theorem for a convex polyhedral set.

Theorem 3.1. A set T *comprising points* X, *representable in the form (3.5), is a convex polyhedral set.*

Proof. 1. Consider a polyhedral set K of points

$$\overline{Y} = (Y, V) = (y_1, y_2, \ldots, y_n, V),$$

defined by the restraints

$$\begin{aligned} (Y, X_i) - V \leqslant 0, \qquad & i=1, 2, \ldots, N_1, \\ (Y, R_i) \leqslant 0, \qquad & i=1, 2, \ldots, N_2. \end{aligned} \tag{3.15}$$

It follows from (3.15) that K is a polyhedral cone with apex at the point $0=(0, 0, \ldots, 0)$. Therefore (see (3.4)), there exist points $\overline{Y}_1, \overline{Y}_2, \ldots, \overline{Y}_N$ such that the cone K can be written as

$$\overline{Y} = (Y, V) = \sum_{i=1}^{N} \gamma_i \overline{Y}_i = \sum_{i=1}^{N} \gamma_i (Y_i, V_i), \quad \gamma_i \geqslant 0. \tag{3.16}$$

With the points $\overline{Y}_1, \overline{Y}_2, \ldots, \overline{Y}_N$ we form a polyhedral set T_1 of points X whose defining restraints have the form

$$(Y_i, X) \leqslant V_i, \qquad i=1, 2, \ldots, N. \tag{3.17}$$

2. We shall show that the polyhedral set T_1 is identical with the given set T. Consider an arbitrary point $X \in \mathrm{T}$. By assumption,

$$X = \sum_{l=1}^{N_1} \alpha_l X_l + \sum_{l=1}^{N_2} \beta_l R_l,$$

where $\alpha_l \geqslant 0$, $\beta_l \geqslant 0$, $\sum_{l=1}^{N_1} \alpha_l = 1$. Hence, for any $i\,(1 \leqslant i \leqslant N)$,

$$(Y_i, X) = \sum_{t=1}^{N_1} \alpha_t (Y_i, X_t) + \sum_{t=1}^{N_2} \beta_t (Y_i, R_t) \leqslant V_i \sum_{t=1}^{N_1} \alpha_t = V_i. \tag{3.18}$$

We have used the inequalities (3.15), which apply for the points $\overline{Y}_i$, and the properties of the coefficients α_t, β_t.

Inequalities (3.18) indicate that $X \in \mathrm{T}_1$. We have thus proved that

$$\mathrm{T} \subset \mathrm{T}_1.$$

3. Let us now establish the inverse. Consider a point $X_0 \in \mathrm{T}_1$ and assume that

$$X_0 \notin \mathrm{T}.$$

According to Lemma 3.3, T is a closed set. The convexity of T is obvious. Hence, the hyperplane separation theorem (see Appendix, Theorem 3.1) applies to this set and to the point $X_0 \notin T$. According to this theorem there exists a vector Λ_0 and a number c_0 so that

$$(\Lambda_0, X_0) > c_0, \quad (\Lambda_0, X) \leqslant c_0 \tag{3.19}$$

for all $X \in T$. Since (3.20) holds, in particular, for all X_i, the point $\bar{\Lambda}_0 = (\Lambda_0, c_0)$ satisfies the first N_1 restraints of the system (3.15).

Let X' be some point in T. In this case, for any l and $\beta \geqslant 0$ the point

$$X_\beta = X' + \beta R_l \in T.$$

Hence,

$$(\Lambda_0, X_\beta) = (\Lambda_0, X') + \beta(\Lambda_0, R_l) \leqslant c_0$$

for any $\beta \geqslant 0$. This is possible only if

$$(\Lambda_0, R_l) \leqslant 0.$$

The point $\bar{\Lambda}_0 = (\Lambda_0, c_0)$ therefore satisfies the last N_2 restraints of (3.15).

Thus, $\bar{\Lambda}_0 \in K$. Hence, applying (3.16), we have

$$\Lambda_0 = \sum_{l=1}^{N} \gamma_l^{(0)} Y_l,$$

$$c_0 = \sum_{l=1}^{N} \gamma_l^{(0)} V_l, \quad \gamma_l^{(0)} \geqslant 0.$$

By assumption, X_0 satisfies (3.17) ($X_0 \in T_1$). Hence

$$(\Lambda_0, X_0) = \sum_{l=1}^{N} \gamma_l^{(0)} (Y_l, X_0) \leqslant \sum_{l=1}^{N} \gamma_l^{(0)} V_l = c_0.$$

This inequality contradicts (3.19). The assumption that $X_0 \notin T$ is therefore false. Thus, any point X_0 of T_1 belongs to T, i. e.,

$$T_1 \subset T.$$

From this relationship and that established at the end of point 2, we have

$$T = T_1.$$

This completes the proof.

3-5. For the sake of brevity, we say that the set T comprising points X of the form (3.5) is spanned (or generated) by the points $X_1, X_2, \ldots, X_{N_1}$ and the vectors $R_1, R_2, \ldots, R_{N_2}$.

We remind the reader that points and vectors are, conceptually, equivalent. Referring to $X_1, \ldots, X_{N_1}$ as points and to $R_1, R_2, \ldots, R_{N_2}$ as vectors we only emphasize the difference in the roles of these elements in spanning T; the set T coincides with the sum of convex combinations of the points $X_1, X_2, \ldots, X_{N_1}$ and nonnegative linear combinations of the vectors $R_1, R_2, \ldots, R_{N_2}$. As we have already indicated, any polyhedral set is spanned by a finite number of points and vectors. On the other hand, by Theorem 3.1, proved above, any set spanned by a finite number of points and vectors is a polyhedral set. Thus, we can introduce an alternative definition of a convex polyhedral set.

A convex polyhedral set is a set spanned by a finite number of points and vectors (a set which is the sum of convex combinations of the given points, and nonnegative linear combinations of the given vectors). We shall refer to this definition as the second definition of a polyhedral set. The definition given in § 1, according to which a polyhedral set is the intersection of a finite number of half-spaces and hyperplanes will be

referred to below as the first definition of a polyhedral set. The equivalence of these two definitions, established by Theorems 2.6 and 3.1, is one of the underlying concepts of the theory of convex polyhedral sets.

Above we discussed two important classes of convex polyhedral sets, i.e., the class of convex polyhedra and the class of convex polyhedral cones. According to the first definition, a convex polyhedron is a bounded set which is the intersection of a finite number of half-spaces and hyperplanes. Since a bounded set may not contain rays, the second definition of a convex polyhedron is formulated as follows: a convex polyhedron is a set comprising all the convex combinations of a finite number of points.

We have seen that a polyhedral set is a convex polyhedral cone if and only if the restraint system of this set can be reduced to the form (3.1), (3.2). Therefore, according to the first definition of a polyhedral set, a convex polyhedral cone may be defined as the intersection of a finite number of hyperplanes and half-spaces, whose boundary hyperplanes intersect at a single point, called the apex of the cone. (We note that the boundary hyperplane of a hyperplane is the hyperplane itself.)

Applying the second definition of a polyhedral set, we say that a convex polyhedral cone with apex at X_0, consists of all nonnegative linear combinations of a finite number of vectors translated to the point X_0.

§ 4. Principal properties of the linear-programming problem

4-1. Consider the general linear-programming problem written in arbitrary form.

Maximize the linear form

$$(C, X)=\sum_{j=1}^{n} c_j x_j \tag{4.1}$$

subject to the conditions

$$(D_i, X)=\sum_{j=1}^{n} d_{ij}x_j \leqslant d_i, \qquad i=1,\ 2,\ \ldots,\ s; \tag{4.2}$$

$$(D_i, X)=\sum_{j=1}^{n} d_{ij}x_j = d_i, \qquad i=s+1,\ \ldots,\ s+t. \tag{4.3}$$

We assume that the restraint system (4.2), (4.3) is consistent. In this case all the points X satisfying (4.2), (4.3) constitute a convex polyhedral set. Let this set be denoted by M. Problem (4.1)-(4.3) thus entails maximizing a linear function (4.1) defined on a polyhedral set M. This set is sometimes called the domain of definition of the linear form, or the polyhedral set of the restraints of the given problem.

In this section we indicate some important properties of the problem (4.1)-(4.3); in the process we shall draw upon the various aspects of the theory of polyhedral sets discussed in previous sections.

We first observe that the rank of the restraint system (4.2), (4.3) (the rank of the system of vectors D_i, $i=1, 2, \ldots, s+t$) can always be taken to equal the dimensionality of the space of points X, i.e., n. To verify this, assume that the rank of system of restraints (4.2), (4.3) is $r<n$. Applying

Lemma 2.1, we have the representation

$$M = M_0 + \Pi_{n-r}, \tag{4.4}$$

where M_0 is a polyhedral set, whose defining system of restraints is of rank n, Π_{n-r} is an $(n-r)$-dimensional subspace. Two cases are possible:

(a) a point $Y \in \Pi_{n-r}$ exists so that

$$(C, Y) = \sum_{i=1}^{n} c_i y_i = \zeta \neq 0;$$

(b) $(C, Y) = 0$ for any $Y \in \Pi_{n-r}$.

It is easily seen that in case (a) the linear form (4.1) is unbounded from above on the set M(on the set of feasible programs of the problem). Indeed, since Π_{n-r} is a subspace, $\lambda Y \in \Pi_{n-r}$ for any λ.

Let $X_\lambda = X + \lambda Y$, where X is any point in M_0. According to the representation (4.4), $X_\lambda \in M$ for any λ. Consider

$$(C, X_\lambda) = (C, X) + \lambda (C, Y) = (C, X) + \lambda \zeta.$$

Obviously, $\sup (C, X_\lambda) = \infty$ for $\zeta \neq 0$, i. e., the linear form of the problem is unbounded in its domain of definition.

For case (b), consider the maximization of the linear form (4.1), subject to the restraints defining the polyhedral set M_0. If this problem is unsolvable, then according to (4.4) the problem (4.1)-(4.3) has no solution either. Let M^* and M_0^* denote the sets of optimal programs with the linear form (4.1) and restraint polyhedral sets M and M_0, respectively. Applying (b), we obtain from (4.4)

$$M^* = M_0^* + \Pi_{n-r}.$$

Thus, to solve the problem (4.1)-(4.3), the polyhedral set M may be replaced by M_0. If the resulting problem is solvable and (b) is satisfied, then solution of this problem yields the optimal program of the initial problem. Otherwise, the initial problem has no solution.

The transition from M to M_0 entails calculating the maximum number of linearly independent vectors in the system D_i, $i = 1, 2, \ldots, s+t$. The subspace Π_{n-r} is obtained by representing the remaining D_i in terms of the linearly independent subsystem (see 2-3).

In this section we shall assume that these transformations have been carried out (if transformations were indeed necessary) so that the rank of the restraint system defining the polyhedral set of the problem is always taken to equal n. The necessity of carrying out these transformations is generally established during the preliminary analysis of the problem (when the first approximation is being computed). This aspect of the problem is dealt with when encountered.

4-2. When analyzing linear-programming problems, special significance is attached to programs corresponding to the vertices of the polyhedral set of restraints. (In the following the reader will often see the validity of this statement.) We shall, therefore, discuss separately the class of these programs.

A program $X = (x_1, \ldots, x_n)$ of the problem (4.1)-(4.3) is called a s u p p o r t p r o g r a m if there are n linearly independent restraints among the relationships (4.2), (4.3), which are reduced by the program to equalities.

It follows from this definition that the concept of a support program is equivalent to a vertex of the polyhedral set defined by the restraints (4.2),

(4.3) (see Theorem 1.4). The number of support programs of a linear-programming problem is, therefore, always finite. When the domain of definition of the linear form of the problem is bounded, i.e., a convex polyhedron, any feasible program of the problem is a convex linear combination of its support programs.

Since the rank of the restraint system (4.2), (4.3) is n, applying Theorem 2.4, we obtain the following result.

Theorem 4.1. (Existence theorem for support programs.) If the set of feasible programs of problem (4.1)-(4.3) is not empty, at least one of the feasible programs is a support program.

We remind the reader that a solution of the linear-programming problem is a program for which the linear form of the problem is conditionally maximized or minimized, depending on the exact statement of the problem.

A problem having at least one solution is said to be solvable.

Let $X_1, X_2, \ldots, X_{N_1}$ consist of all the support programs of the problem (4.1)-(4.3); $R_1, R_2, \ldots, R_{N_2}$ are the direction vectors of all the unbounded edges of the polyhedral set M. In this case, according to Theorem 2.2, the polyhedral set of restraints of the problem (4.1)-(4.3) comprises points of the form X

$$X=\sum_{i=1}^{N_1} \alpha_i X_i+\sum_{i=1}^{N_2} \beta_i R_i, \tag{4.5}$$

where $\alpha_i \geqslant 0,\ \beta_i \geqslant 0,\ \sum_{i=1}^{N_1} \alpha_i=1.$

Assume that the problem (4.1)-(4.3) is solvable. Then it can be easily verified that for any i, $1 \leqslant i \leqslant N_2$,

$$(C, R_i) \leqslant 0. \tag{4.6}$$

Indeed, if for some $i=i'$

$$(C, R_{i'})=\delta>0,$$

then, taking $X(\beta)=X_1+\beta R_{i'}$, we obtain

$$\lim_{\beta \rightarrow \infty}(C, X(\beta))=\infty. \tag{4.7}$$

But, for any $\beta \geqslant 0$ the vector $X(\beta)$ is a program of the problem (4.1)-(4.3). Hence, (4.7) indicates that this problem is unsolvable (the linear form of the problem is unbounded from above on the set of feasible programs). Thus, for solvable problems, (4.6) applies for any vector R_i.

Assume now that X^* is a solution of the problem (4.1)-(4.3). Applying (4.5), which is valid for any feasible program of the problem, we have

$$\begin{gathered}X^*=\sum_{i=1}^{N_1} \alpha_i^* X_i+\sum_{i=1}^{N_2} \beta_i^* R_i, \\ \alpha_i^* \geqslant 0, \quad \beta_i^* \geqslant 0, \quad \sum_{i=1}^{N_1} \alpha_i^*=1.\end{gathered} \tag{4.8}$$

We now choose a subscript $i=i'$ so that $\alpha_{i'}^*>0$. We express $X_{i'}$ from (4.8):

$$X_{i'}=\frac{1}{\alpha_{i'}^*}\left[X^*-\sum_{i \neq i'} \alpha_i^* X_i-\sum_{i=1}^{N_2} \beta_i^* R_i\right].$$

Taking the scalar product of both sides of this equality with the vector C,

we obtain

$$(C, X_{l'}) = \frac{1}{\alpha_{l'}^{*}}\left[(C, X^{*}) - \sum_{l \neq l'} \alpha_l^{*}(C, X_l) - \sum_{l=1}^{N_2} \beta_l^{*}(C, R_l)\right] \geqslant$$
$$\geqslant \frac{1}{\alpha_{l'}^{*}}\left[(C, X^{*}) - (C, X^{*}) \sum_{l \neq l'} \alpha_l^{*} - \sum_{l=1}^{N_2} \beta_l^{*}(C, R_l)\right] \geqslant$$
$$\geqslant \frac{1}{\alpha_{l'}^{*}}\left[(C, X^{*}) - (C, X^{*}) \sum_{l \neq l'} \alpha_l^{*}\right] =$$
$$= (C, X^{*}) \frac{1}{\alpha_{l'}^{*}}\left(1 - \sum_{l \neq l'} \alpha_l^{*}\right) = (C, X^{*}).$$

The first inequality follows from the fact that X^{*} solves the problem; the second inequality follows from (4.6); the last equality follows from the condition

$$\sum_{l=1}^{N_1} \alpha_l^{*} = 1.$$

Thus,

$$(C, X_{l'}) \geqslant (C, X^{*}).$$

On the other hand, since X^{*} is a solution of the problem we have

$$(C, X_{l'}) \leqslant (C, X^{*}).$$

Comparing the last two inequalities, we obtain

$$(C, X_{l'}) = (C, X^{*}), \tag{4.9}$$

where, by assumption, $X_{l'}$ is a support program of the problem (4.1)–(4.3).

A solution providing a support program of a linear-programming problem will be called a support solution of this problem. The relationship (4.9) is equivalent to the following proposition, which is highly important in linear programming.

Theorem 4.2. (Existence theorem for support solutions.) Any solvable linear-programming problem with a restraint system of rank n has at least one support solution.

4-3. Theorem 4.2 indicates the following method for solving linear-programming problems. Compute all the support programs of the problem. This can be done by investigating C_s^{n-t} systems of linear equations each containing t equations (4.3), and $n-t$ equations satisfying the conditions (4.2). Then, compute the value of the linear form (4.1) for each of the resulting support programs, whose number is finite. By Theorem 4.2, if the problem is solvable, the support program corresponding to the highest value of the linear form solves it. However, this method is impracticable if $n-t$ and $s \gg n-t$ are fairly large (as is generally the case in applied problems). Let, for example, $t=0$, $s=2n$. Applying Stirling's formula, we have

$$C_{2n}^{n} \approx \frac{1}{\sqrt{\pi n}} 2^{2n}.$$

Investigating n linear equations in n unknowns (determining their solutions or detecting linear dependence in the given system of vectors) requires about n^3 operations.

Hence, $\frac{n^3}{\sqrt{n}} 2^{2n}$ operations will be required to establish all the support programs of the problem. Assuming the problem to be programmed for a

computer performing 10^5 operations per second, the optimal program for $n=25$ will be found after approximately 10^6 years.

Even for comparatively small problems ($n=25$), this method involves an astronomical number of computations and is obviously of no practical significance. Nevertheless, in spite of the unsuitability of the method based on evaluating all the vertices of the polyhedral restraint set, the concept of vertex evaluation is quite useful. All the methods (apart from the iterative methods) of linear programming entail, to a certain extent, evaluation of a sequence of vertices of some polyhedral set (not necessarily the polyhedral set of restraints of the problem in question). This evaluation, however, is carried out in such a way that only a small part of all the vertices available need be considered to solve the problem. An essential reduction in alternative programs to be considered is achieved with the aid of two properties which must be characteristic of any practicable method of evaluation:

(a) ordered evaluation, i. e., arranging the vertices so that a "worse" vertex is never preceded by a "better" vertex (the concepts of "worse" and "better" are inherent of the method);

(b) availability of a criterion making it possible to decide, without evaluating all the vertices, that the "best" vertex has been found (the vertex giving the solution of the problem). Each linear-programming problem has its characteristic method of ordered evaluation and criterion.

4-4. If a linear-programming problem has a unique solution, then, by Theorem 4.2, this is a support solution. Assume that the problem has several solutions. Consider one of the solutions, say X^*. The vector X^* can be represented in the form (4.8). In deriving (4.9) we showed that if $\alpha_i^* > 0, X_i$ solves the problem (4.1)–(4.3). Analogously, we may verify that $(C, R_i)=0$, for $\beta_i^* > 0$. Any solution X^* of the problem (4.1)–(4.3) can be represented as

$$X^* = \sum_{\lambda=1}^{n_1^*} \alpha_{i_\lambda}^* X_{i_\lambda} + \sum_{\lambda=1}^{n_2^*} \beta_{i_\lambda}^* R_{i_\lambda}, \tag{4.10}$$

where $\alpha_{i_\lambda}^* > 0,\ \lambda=1, 2, \ldots, n_1^*,\ \beta_{i_\lambda}^* > 0,\ \lambda=1, 2, \ldots, n_2^*,\ \sum_{\lambda=1}^{n_1^*} \alpha_{i_\lambda} = 1.$ Here

$$(C, X^*) = (C, X_{i_\lambda}), \qquad \lambda=1, 2, \ldots, n_1^*; \tag{4.11}$$

$$(C, R_{i_\lambda}) = 0, \qquad \lambda=1, 2, \ldots, n_2^*. \tag{4.12}$$

Let us isolate all the support solutions of the problem $X_{i_\lambda}, \lambda=1, 2, \ldots, n_1$, and all the vectors $R_{i_\lambda}, \lambda=1, 2, \ldots, n_2$, satisfying (4.12). Consider the polyhedral set M^* spanned by the points X_{i_λ} and the vectors R_{i_λ}, i. e., the set of points X of the form

$$X = \sum_{\lambda=1}^{n_1} \alpha_{i_\lambda} X_{i_\lambda} + \sum_{\lambda=1}^{n_2} \beta_{i_\lambda} R_{i_\lambda}, \tag{4.13}$$

$$\alpha_{i_\lambda} \geqslant 0,\quad \lambda=1, 2, \ldots, n_1,\qquad \beta_{i_\lambda} \geqslant 0,\quad \lambda=1, 2, \ldots, n_2,$$

$$\sum_{\lambda=1}^{n_1} \alpha_{i_\lambda} = 1.$$

It follows from (4.10)–(4.12) that any solution X^* of the problem (4.1)–(4.3) belongs to M^*. Since the support programs X_{i_λ} and the vectors R_{i_λ} satisfy conditions (4.11) and (4.12), respectively, any point in M^* solves the problem in question. Thus, all the solutions of the problem (4.1)–(4.3) constitute a polyhedral set M^* comprising points written in the form (4.13). We have thus proved the following proposition:

Theorem 4.3. All the solutions of the linear-programming problem constitute a polyhedral set spanned by the support solutions of the problem and those direction vectors R_i of the unbounded edges of M which satisfy the equality (4.12).

From Theorem 4.3 it follows, in particular, that the linear-programming problem is either uniquely solvable or has an infinite number of solutions. In the latter case, however, all the solutions can be expressed in terms of a finite number of vectors X_{i_λ} and R_{i_λ}.

4-5. As we have indicated in Chapter 1, the linear-programming problem may be unsolvable because the restraint system is inconsistent (the set of feasible programs is empty), or the linear form of the problem is not bounded (from above or below, depending on the statement of the problem) on the set of feasible programs. The following theorem shows that there are no other reasons which may render the linear-programming problem unsolvable.

Theorem 4.4. (Solvability theorem of linear programming.) If the set of feasible programs of a linear-programming problem is nonempty and the linear form of the problem is bounded from above on this set (the case of maximization), the problem in question is solvable, i.e., has at least one solution.

Proof. Since the set of feasible programs is not empty, it is a polyhedral set and as such comprises points X of the form

$$X=\sum_{i=1}^{k_1}\alpha_i X_i+\sum_{i=1}^{k_2}\beta_i R_i, \tag{4.14}$$

where all $\alpha_i \geqslant 0, \beta_i \geqslant 0, \sum_{i=1}^{k_1}\alpha_i=1$, X_i, $i=1, 2, \ldots, k_1$ are some feasible programs of the problem; R_i, $i=1, 2, \ldots, k_2$ is some system of vectors.

Observe that we apply here the general representation theorem of polyhedral sets (Theorem 2.6), which does not presuppose equality of rank of the system of restraints defining this set with the dimensionality n of the space of points X. The linear form of the problem, by assumption, is bounded from above. Therefore, for any vector R_i,

$$(C, R_i)\leqslant 0 \tag{4.15}$$

(see (4.6)).

Let X be an arbitrary program of this problem. From (4.14), this point can be written as

$$X=X'+X'',$$

where $X'=\sum_{i=1}^{k_1}\alpha_i X_i$; $X''=\sum_{i=1}^{k_2}\beta_i R_i$. Applying (4.15) and noting that β_i are nonnegative, we obtain

$$(C, X)=(C, X')+(C, X'')\leqslant (C, X'). \tag{4.16}$$

We choose $X_{i'}$ from the condition

$$(C, X_{i'})=\max_{1\leqslant i\leqslant k_1}(C, X_i).$$

In this case, applying the properties of the coefficients α_i, we have

$$(C, X')\leqslant (C, X_{i'}). \tag{4.17}$$

Comparing inequalities (4.16) and (4.17), we obtain

$$(C, X) \leqslant (C, X_{l'})$$

for any feasible program X of the problem in question.

This inequality indicates that the program $X_{l'}$ solves the problem. This completes the proof.

Generally, Theorem 4.4 applies only to linear-programming problems. If we consider a problem involving maximization of a linear function defined on some convex closed set D, which is not a polyhedral set, Theorem 4.4 does not generally apply.

For example, maximize the form

$$x_2 \tag{4.18}$$

subject to the conditions

$$x_1 x_2 \leqslant -1, \quad x_1 \geqslant 0. \tag{4.19}$$

The restraints (4.19) define in the (x_1, x_2)-plane a convex closed domain D bounded by the branch of the hyperbola $x_1 x_2 = -1$ lying in the fourth quadrant.

It is easily observed (Figure 2.3) that

$$\sup_{(x_1, x_2) \in D} x_2 = 0.$$

Here $x_2 \leqslant 0$, and the point $\left(\frac{1}{\varepsilon}, -\varepsilon\right) \in D$ for any $\varepsilon > 0$. On the other hand, no point $(x_1, 0) \in D$ exists. The upper bound of the function (4.18) defined on D is therefore not attained on any point in D.

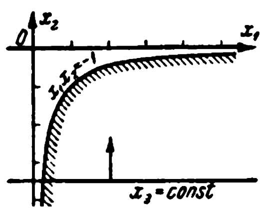


FIGURE 2.3

Thus, all the requirements of Theorem 4.4 are satisfied in this problem. Nevertheless, the problem has no solution. Examples of certain discontinuous nonlinear functions defined on convex polyhedral sets can also be given for which Theorem 4.4 does not hold.

We thus see that the assumptions of linearity of the function to be optimized, and the polyhedrality of its domain of definition required in Theorem 4.4, are most significant.

4-6. Until now we have considered the linear-programming problem written in arbitrary form. In describing methods of linear programming, we find it more convenient to deal with the canonical form of the problem (see Chapter 1, § 5). We now proceed with the discussion of the general linear-programming problem given in canonical form.

Maximize the linear form

$$\sum_{j=1}^{n} c_j x_j \tag{4.20}$$

subject to the conditions

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \qquad i = 1, 2, \ldots, m; \tag{4.21}$$

$$x_j \geqslant 0, \qquad j=1, 2, \ldots, n. \tag{4.22}$$
$$m<n$$

We remind the reader that the vectors $A_j=(a_{1j}, a_{2j}, \ldots, a_{mj})^T$, $j=1,2,\ldots,n$, and the vector $B=(b_1, b_2, \ldots, b_m)^T$ are generally called the restraint vectors and the constraint vector, respectively. Equalities (4.21) are sometimes written more conveniently in vector form:

$$\sum_{j=1}^{n} A_j x_j = B. \tag{4.21'}$$

Problem (4.20)–(4.22) is a particular case of the problem (4.1)–(4.3) for $t=m$, $s=n$,

$$D_i=\begin{cases} (\underbrace{0, 0, \ldots, 0, -1}_{i}, 0, \ldots, 0), & i=1, 2, \ldots, n, \\ (a_{i-n,1}, a_{i-n,2}, \ldots, a_{i-n,n}), & i=n+1, n+2, \ldots \\ & \ldots, n+m, \end{cases}$$

$$d_i=\begin{cases} 0, & i=1, 2, \ldots, n, \\ b_{i-n}, & i=n+1, n+2, \ldots, n+m. \end{cases}$$

Since the determinant of the vectors $D_1, D_2, \ldots, D_n$ does not vanish (it is equal to $(-1)^n$), the restraint system (4.21), (4.22) is always of rank n. We shall assume that all the equalities of system (4.21) are linearly independent. This assumption does not limit the scope of our discussion, since otherwise some of the equations (4.21) could be omitted without modifying the polyhedral set of restraints of the problem.

Let us find a support program for the problem (4.20)–(4.22).

Let $X=(x_1, x_2, \ldots, x_n)$ be a feasible program of the problem (4.20)–(4.22). According to the definition of a support program, there exist n linearly independent vectors among the D_i satisfying

$$(D_i, X)=d_i. \tag{4.23}$$

The vectors D_{i+n}, $i=1,2,\ldots,m$ are linearly independent and satisfy (4.23).

Therefore, if (and only if) X is a support program, there exist among the vectors D_i, $1 \leqslant i \leqslant n$, $n-m$ vectors D_{i_λ}, $1 \leqslant \lambda \leqslant n-m$, which satisfy (4.23) and constitute, together with the vectors D_{i+n}, $1 \leqslant i \leqslant m$, a linearly independent system. This indicates that the determinant of the matrix whose rows are the vectors

$$D_{i_\lambda}=-e_{i_\lambda}^T, \quad 1 \leqslant \lambda \leqslant n-m, \quad D_{i+n}=(a_{i1}, a_{i2}, \ldots, a_{in}),$$
$$1 \leqslant i \leqslant m,$$

does not vanish. Expanding this determinant in the elements of the first $n-m$ rows, we obtain the condition

$$|(A_{j_1}, A_{j_2}, \ldots, A_{j_m})| \neq 0. \tag{4.24}$$

Here $(A_{j_1}, A_{j_2}, \ldots, A_{j_m})$ denotes the matrix consisting of the restraint vectors A_{j_μ}. The set of subscripts $I_X=\{j_1, j_2, \ldots, j_m\}$ is obtained from the complete set $1, 2, \ldots, n$ by neglecting the subscripts i_λ, $1 \leqslant \lambda \leqslant n-m$. Since $x_{i_\lambda}=0$, $1 \leqslant \lambda \leqslant n-m$, then, for $x_j>0, j \in I_X$.

Hence, condition (4.24) can be expressed as follows:

There exists a linearly independent system of m restraint vectors A_j, $j \in I_X=(j_1, j_2, \ldots, j_m)$, containing all those A_j for which $x_j>0$.

This requirement is, obviously, equivalent to the assumption of linear

independence of the system of vectors A_j corresponding to the positive components of the program X.

Thus, a support program for a canonical linear-programming problem can be defined as follows:

A program $X=(x_1, x_2, \ldots, x_n)$ of the problem (4.20)–(4.22) is said to be a *support program* if the restraint vectors corresponding to its positive components are linearly independent.

A system of m linearly independent restraint vectors comprising all the A_j, for which $x_j>0$, is generally called the *basis* of the support program $X=(x_1, x_2, \ldots, x_n)$. The components of a support program connected with the basis vectors are sometimes called the *basis components* of this program.

When solving linear-programming problems we have to deal not only with the vertices of the corresponding polyhedral sets, but also with the edges of these sets. The properties of edges of polyhedral sets were discussed in 1-4 (see Theorem 1.5) with reference to a linear-programming problem written in arbitrary form. We shall now try to apply these properties to the problem (4.20)–(4.22). Since the rank of the restraint system (4.21), (4.22) is n, any edge of the polyhedral set of the problem (4.20)–(4.22) is either a ray or a segment. We know that the end points of an edge of a polyhedral set are vertices of this set (support programs of the corresponding linear-programming problem).

The equation of any edge Γ of the polyhedral set of restraints of the problem (4.20)–(4.22) can be written in the form

$$\overline{X}=X+\lambda e_0, \tag{4.25}$$

where X is a support program of the problem, and e_0 is the direction vector of the edge. The parameter λ ranges from 0 to $\lambda_0<\infty$ if Γ is a segment and from 0 to ∞ if Γ is a ray.

Let E_X be the set of subscripts i for which the equality (4.23) is satisfied. Obviously, $n+i\in E_X$ for $1\leqslant i\leqslant m$. Since X is a support program, the rank of the system of vectors D_i, $i\in E_X$, is n. According to the definition of an edge of a polyhedral set, the nonzero vector e_0 (the direction vector of the edge Γ) satisfies the following system of homogeneous equations:

$$(D_i, e_0)=0, \quad i=i_1, i_2, \ldots, i_{n-m-1}, \quad n+1, n+2, \ldots, n+m,$$
$$1\leqslant i_\lambda\leqslant n, \quad i_\lambda\in E_X, \quad \lambda=1, 2, \ldots, n-m-1.$$

The vectors $D_{i_1}, \ldots, D_{i_{n-m-1}}, D_{n+1}, \ldots, D_{n+m}$ are linearly independent.

Let $I_e=\{j_1, j_2, \ldots, j_m, j_{m+1}\}$ be the set of subscripts obtained from the set $1, 2, \ldots, n$ where the subscripts i_λ, $1\leqslant\lambda\leqslant n-m-1$ are neglected. Remembering the specific features of the vectors D_i for $i=1, 2, \ldots, n$, we formulate the preceding conditions to be satisfied by the vector e_0 as follows:

(a) the vector $e_0=(e_{01}, e_{02}, \ldots, e_{0n})$ satisfies the equalities

$$\sum_{j=1}^{n} a_{ij}e_{0j}=0, \quad i=1, 2, \ldots, m;$$

(b) for $j\notin I_e$, $e_{0j}=0$;

(c) among the restraint vectors A_j, $j\in I_e$ there exist m linearly independent vectors.

Since the subscripts $i_\lambda\in E_X$, $\lambda=1, 2, \ldots, n-m-1$, we have $I_X\subset I_e$.

The basis of the support program X (the end point of the edge Γ) is thus obtained from the system of vectors A_j, $j\in I_e$ by eliminating one of the vectors of this system. Assume that the vector $A_{j_{m+1}}$ is to be eliminated.

In this case $I_X = \{j_1, j_2, \ldots, j_m\}$. If the basis of the support program X is given, the edge Γ issuing from the vertex X is uniquely determined by the restraint vector $A_{j_{m+1}}$. Let us expand the vector $A_{j_{m+1}}$ in terms of basis vectors of the program X:

$$A_{j_{m+1}} = \sum_{\alpha=1}^{m} A_{j_\alpha} x_\alpha^{(\Gamma)}.$$

The coefficient $x_\alpha^{(\Gamma)}$ can be used to establish whether the edge Γ is bounded. From the conditions determining the vector e_0 we have

$$e_{0j_\alpha} = -e_{0j_{m+1}} x_\alpha^{(\Gamma)}, \qquad \alpha = 1, 2, \ldots, m.$$

Substituting the above in (4.25) and taking into account that $e_{0j} = 0$ for $j \notin I_e$ and $x_j = 0$ for $j \notin I_X$, we obtain

$$x_j(\lambda) = \begin{cases} x_{j_\alpha} - \lambda x_\alpha^{(\Gamma)}, & j = j_\alpha, \quad \alpha = 1, 2, \ldots, m, \\ \lambda, & j = j_{m+1}, \\ 0, & j \notin I_e. \end{cases} \tag{4.26}$$

Here $X(\lambda) = (x_1(\lambda), \ldots, x_n(\lambda))$ is an arbitrary point of edge Γ. Since the vector $X(\lambda)$ satisfies the restraints (4.21) for any λ, this vector is a feasible program of the problem if and only if

$$x_{j_\alpha}(\lambda) = x_{j_\alpha} - \lambda x_\alpha^{(\Gamma)} \geqslant 0, \quad \alpha = 1, 2, \ldots, m.$$

If $x_\alpha^{(\Gamma)} \leqslant 0$ for $\alpha = 1, 2, \ldots, m$, these inequalities are satisfied for any $\lambda \geqslant 0$ and, consequently, Γ is an unbounded edge (a ray issuing from the vertex X). If, however, $x_\alpha^{(\Gamma)} > 0$ for some Γ, the edge α' is a segment, since λ cannot exceed the value λ_0, where

$$\lambda_0 = \min_{x_\alpha^{(\Gamma)} > 0} \frac{x_{j_\alpha}}{x_\alpha^{(\Gamma)}} < \infty.$$

The program $X(\lambda_0)$ gives the second end point of the edge Γ. As we have shown previously, $X(\lambda_0)$ is a support program of the problem. It follows from (4.26) that the basis of the program $X(\lambda_0)$ is formed from the basis of the program X by replacing one of the vectors A_j, $j \in I_X$, by the vector $A_{j_{m+1}}$. The vertices of a polyhedral set which are end points of the same edge are, naturally, called *adjoining vertices*. The corresponding support programs are, therefore, sometimes called adjoining support programs. One of the most effective techniques of linear programming, the simplex method, deals with evaluation of the adjoining support programs of the problem.

4-7. When describing various techniques of linear programming, it is often convenient to isolate the class of problems comprising the so-called nondegenerate linear-programming problems. We now give the relevant definitions.

A support program of a linear-programming problem (4.1)–(4.3) is said to be *nondegenerate* if the number of restraints in the system (4.2), (4.3), reduced by this program to equalities, is n. All these relationships should obviously be linearly independent. If the support program does not satisfy the requirement of nondegeneracy, i. e., if it reduces to equalities more than n restraints from (4.2), (4.3), it is called a *degenerate program*.

A linear-programming problem is called *nondegenerate*, if all its

support programs are nondegenerate. A problem with at least one degenerate support program is called a *degenerate problem*.

These definitions are obviously meaningful only if the problem in question has support programs, i.e., if the restraint system (4.2), (4.3) is of rank n.

If the support program X of problem (4.1)–(4.3) is nondegenerate, we may easily determine the number of edges issuing from the vertex X of the polyhedral restraint set.

Assume that
$$(D_i\, X) = d_i$$
for $i = i_1, i_2, \ldots, i_{n-t}, s+1, s+2, \ldots, s+t, 1 \leqslant i_\lambda \leqslant s$. Let the set of these subscripts be E_X. By assumption,
$$(D_i, X) < d_i, \text{ if } \quad i \notin E_X \tag{4.27}$$
(X is a nondegenerate program). Let $e^{(\alpha)} = (e_{\alpha 1}, \ldots, e_{\alpha n}), 1 \leqslant \alpha \leqslant n-t$, be a vector whose components are solutions of the equations
$$(D_i, e^{(\alpha)}) = \begin{cases} 0, & i \in E_X, \quad i \neq i_\alpha, \\ -1, & i = i_\alpha. \end{cases} \tag{4.28}$$
Consider the set Γ_α of points of the polyhedral set M written in the form
$$X + \lambda e^{(\alpha)}, \quad \lambda \geqslant 0.$$
From equations (4.28) and inequalities (4.27) it follows that Γ_α contains a certain segment, i.e., it is a one-dimensional set. Further, applying the first $n-1$ equations of system (4.28) we conclude that Γ_α is an edge of M with $e^{(\alpha)}$ as its direction vector.

On the other hand, the direction vector e of any edge Γ issuing from the vertex X must satisfy the equations
$$(D_i, e) = 0, \quad i \in E_X, \quad i \neq i_\alpha,$$
for some α, $1 \leqslant \alpha \leqslant n-t$. The vector e should therefore be parallel to one of the vectors $e^{(\alpha)}$ which, consequently, can be taken as the direction vector of the edge Γ. The edge Γ thus coincides with one of the edges Γ_α.

Thus, precisely $n-t$ edges of the polyhedral set M issue from each vertex X, which is a nondegenerate program of the problem (4.1)–(4.3). The direction vector of any such edge can be determined from equations (4.28) for some α, $1 \leqslant \alpha \leqslant n-t$. If the support program X is degenerate, the number of edges of X issuing from the vertex M need not equal $n-t$. This number may be either smaller or greater than $n-t$ (see Exercise 12).

Geometrically, nondegeneracy of a support program X means that exactly n boundary hyperplanes of the polyhedral set M pass through the vertex X. If the problem (4.1)–(4.3) is degenerate an analogous property is characteristic of all the faces of M (and not only of the zero-dimensional ones).

Theorem 4.5. If the problem (4.1)-(4.3) is nondegenerate, then through any q-dimensional face of the polyhedral set M pass exactly $n-q$ boundary hyperplanes of M (all these are obviously linearly independent).

Proof. Let M_q be any q-dimensional face of M. The face M_q is a convex polyhedral set whose defining system of restraints is obtained from (4.2), (4.3), with some of the inequalities in (4.2) replaced by equalities. All the inequalities appearing in the restraint system of M_q can be considered as

imposing nonrigid constraints on M_q. Therefore, according to Theorem 1.2, the rank of the equality restraints in the defining system of M_q is $n-q$.

Let X_q be an arbitrary vertex of the polyhedral set M_q. The existence of this vertex follows from the fact that the rank of the restraint system defining M_q is n (equal to the rank of the restraint system of the polyhedral set M). It is easy to verify that X_q is a vertex of the polyhedral set M (see point 2 in the proof of Theorem 2.2). By assumption X_q is a nondegenerate program. Accordingly, all restraints (4.2), (4.3) reduced by this program to equalities should be linearly independent. In particular this also applies to the equality restraints of the polyhedral set M_q. Since the rank of these restraints is $n-q$, their number also equals $n-q$. This completes the proof.

Consider now a canonical linear-programming problem (4.20)–(4.22). Let X be a support program of this problem. If the restraint vectors A_j, $j \in I_X$, constitute the basis of program X, then $x_j = 0$ for $j \notin I_X$. This indicates that the program X reduces n linearly independent restraints of the defining system of the polyhedral set M to equalities

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \quad i = 1, 2, \ldots, m,$$
$$x_j = 0, \quad j \notin I_X = \{j_1, j_2, \ldots, j_m\}.$$

The requirement of nondegeneracy of the program X reduces all the other restraints of the problem to strict inequalities. Hence, a program X is nondegenerate if and only if

$$x_j > 0 \text{ for } j \notin I_X.$$

Thus, the nondegeneracy of a support program of a canonical linear-programming problem can be defined as follows:

A support program X of the problem (4.20)–(4.22) is said to be *nondegenerate* if all its components corresponding to the basis vectors (basis components) are positive ($x_j > 0$ for $j \in I_X$). Observe that the basis of a nondegenerate support program is determined uniquely as the system of vectors corresponding to the positive components of the program. A degenerate support program may have several bases.

§ 5. The geometry of the linear-programming problem

5-1. As we saw in Chapter 1, geometrical considerations are very useful in the analysis of linear-programming problems. In the following the reader will see that geometrical analogies make the techniques of linear programming much more lucid and constitute the heuristic basis of the various theorems in linear programming. A new approach to the solution of linear-programming problems often arises from elementary geometrical considerations. In this section we shall deal with two geometrical interpretations of the general linear-programming problem with any number of restraints and unknowns. We suggest that the reader, before carrying on with this section, review Chapter 1, § 6 where the two interpretations are considered with reference to two- and three-dimensional cases. The corresponding drawings (Figures 1.1–1.5) given there serve as good illustrations of the somewhat abstract discussion that follows.

5-2. We start with the first geometrical interpretation of the linear-programming problem. Consider the general linear-programming problem (4.1)–(4.3) written in arbitrary form. The restraints (4.2), (4.3) define a convex polyhedral set M in the n-dimensional space of points $X=(x_1, x_2, \ldots, x_n)$ (always assuming the restraints to be consistent). The polyhedral set M may be considered as the intersection of the half-spaces

$$(D_i, X) \leqslant d_i, \quad i=1, 2, \ldots, s$$

and the hyperplanes

$$(D_i, X) = d_i, \quad i=s+1, s+2, \ldots, s+t.$$

The boundary of M consists of parts of the boundary hyperplanes $(D_i, X)=d_i$, $i=1, 2, \ldots, s+t$. Observe that, in general, not all the boundary hyperplanes are part of the boundary of M. The hyperplanes defined by the inequality restraints (4.2) need not have common points with M. Thus the restraints (4.2) defining such hyperplanes could be omitted without affecting the polyhedral set M; however, it is most complicated to find these restraints analytically.

The dimensionality ϱ of the polyhedral set M is not greater than $n-t$, where t is the number of equality restraints (4.3) which are assumed to be linearly independent. If all the restraints (4.2) are nonrigid constraints of M, then $\varrho=n-t$. Therefore, translating the origin of coordinates to some point of intersection of the hyperplanes

$$(D_i, X) = d_i, \qquad i=s+1, \ldots, s+t,$$

we may consider M in an $(n-t)$-dimensional subspace of the principal space. Analytically, this can be achieved by expressing any t variables from equations (4.3) in terms of the other $n-t$ variables and subsequently eliminating the former from the inequalities (4.2). The space containing the set M thus becomes $(n-t)$-dimensional. This technique was already employed in Chapter 1, § 6. The linear form (4.1) of the problem defines, in the n-dimensional space, a system of parallel hyperplanes

$$(C, X) = \lambda, \quad -\infty < \lambda < \infty.$$

Each of these hyperplanes will be called the linear-form hyperplane of the problem. The linear-form coefficients c_j, $j=1, 2, \ldots, n$ constitute a vector $C=(c_1, c_2, \ldots, c_n)$ orthogonal to the family of hyperplanes. The vector C points in the direction of increasing linear form. With fixed λ, the linear-form hyperplane generates two half-spaces. One of these half-spaces containing the point $X+C$ (the point X belongs to the hyperplane) will be called the upper half-space, while the other half-space will be called the lower half-space. The equation of the upper half-space has the form

$$(C, X) \geqslant \lambda,$$

and the lower half-space is defined by the equation

$$(C, X) \leqslant \lambda.$$

For $\lambda=\lambda_0$, let the linear-form hyperplane intersect the polyhedral set of restraints M. The linear form has the same value at all the points of intersection. Translating this hyperplane parallel to itself along the vector C (in the direction of increasing linear form (4.1)), it may happen that further translation will cause the hyperplane and the set M to be disjoint.

Let this limiting position of the hyperplane correspond to $\lambda=\lambda_1$, i. e., the corresponding hyperplane has the equation

$$(C, X)=\lambda_1. \tag{5.1}$$

In this case the polyhedral restraint set of M is located in the lower half-space of hyperplane (5.1). At each of the points belonging to both M and the hyperplane (5.1) (the set of these points is a priori nonempty) the linear form attains the extremum λ_1. The intersection of M and the hyperplane (5.1) defines a polyhedral set M^* of solutions of the problem in question.

If the problem has a unique solution, M^* comprises only one single point, the vertex of M. In the general case, M^* is some convex polyhedral ϱ^*-dimensional set, so that

$$0 \leqslant \varrho^* \leqslant \varrho.$$

Observe that $\varrho^*=\varrho$ only if M belongs to one of the hyperplanes of the family (4.29).

Until now we have assumed the existence of the limiting position of the linear-form hyperplane, as defined by the equality (5.1). This assumption is obviously justified only for a bounded M (if M is a convex polyhedron). If, however, M is an unbounded convex polyhedral set, then even if the linear-form hyperplane is displaced an arbitrarily large distance in the direction specified by the vector C, it may still intersect M. This means that the linear form of the problem is unbounded on the set M, i. e., the given problem is unsolvable. It must be kept in mind that the fact that M is not bounded is not a sufficient condition for the unsolvability of a problem. For some values of the vector C the problem with an unbounded polyhedral restraint set is solvable (the linear-form hyperplane has a finite limit position), and for others it is unsolvable (there exists no finite limit position of the linear-form hyperplane).

Sometimes an erroneous statement of the problem results in inconsistent conditions (4.2), (4.3). Geometrically this corresponds to the case when the domain of definition of the linear form degenerates to an empty set.

5-3. The first geometrical interpretation of the linear-programming problem applies equally well to any representation of the problem. The second geometrical interpretation, which we give below, is suitable only for the canonical form of the problem.

Consider a general linear-programming problem in canonical form (problem (4.20)–(4.22)).

We introduce new variables $u_1, u_2, \ldots, u_m, u_{m+1}$ so that

$$\left.\begin{aligned} u_i &= \sum_{j=1}^{n} a_{ij}x_j, \qquad i=1, 2, \ldots, m; \\ u_{m+1} &= \sum_{j=1}^{n} c_jx_j. \end{aligned}\right\} \tag{5.2}$$

Relationships (5.2) define a transformation of the n-dimensional space of points $X=(x_1, x_2, \ldots, x_n)$ to the $(m+1)$-dimensional space of points $U=(u_1, u_2, \ldots, u_{m+1})$. Let $\overline{A}_j$ denote the $(m+1)$-dimensional column vector with components $a_{1j}, a_{2j}, \ldots, a_{mj}$, i. e.,

$$\overline{A}_j=(a_{1j}, a_{2j}, \ldots, a_{mj}, c_j)^T.$$

The vector $\overline{A}_j$ whose first m components coincide with the components of the restraint vector A_j was called in Chapter 1 the augmented restraint vector.

Relationships (5.2) can be rewritten in vector form

$$U=\sum_{j=1}^{n}\bar{A}_j x_j. \tag{5.3}$$

By definition, the set of points U representable in the form (5.3) with $x_j\geqslant 0$, $j=1, 2, \ldots, n$, is a convex polyhedral cone (see the second definition of a polyhedral cone in 3-5). Denote this cone by K. The cone K is spanned by the augmented restraint vectors $\bar{A}_j$; the apex of K is located at the origin. It follows from (5.3) that K is the image of the positive orthant of the space of points X in the $(m+1)$-dimensional space of points U.

Now, let a point X satisfy the equalities (4.21). Then, according to (5.3), the image of this point in the $(m+1)$-dimensional space in question is the point

$$U_X=(b_1, b_2, \ldots, b_m, L(X)),$$

where $L(X)=\sum_{j=1}^{n} c_j x_j$. On the other hand, for any λ, among the solutions of the system (4.21), there exists a solution X so that $\lambda=L(X)=\sum_{j=1}^{n} c_j x_j$. (We assume here that the vectors C and $(a_{i1}, a_{i2}, \ldots, a_{in})$, $i=1, 2, \ldots, m$ are linearly independent, and $m<n$.)

Hence, (5.3) transforms all the solutions of the system (4.21) to the line Q whose equation is

$$U=\bar{B}+\lambda e_{m+1}, \quad -\infty<\lambda<\infty,$$

where $\bar{B}=(b_1, b_2, \ldots, b_n, 0)$, $e_{m+1}=(0, 0, \ldots, 0, 1)$. The line Q passes through the point $\bar{B}$ and is parallel to the Ou_{m+1}-axis.

Since the feasible programs of the problem (4.20)–(4.22) must satisfy the equalities (4.21) as well as the inequalities (4.22), the transformation (5.3) maps the polyhedral restraint set M onto the intersection of the cone K and the line Q.

Assume that M is nonempty. The cone K and the line Q then have common points. Let the intersection of the line Q and the cone K be denoted by Q_k. The intersection of convex sets is convex. The dimensionality of the set Q_k is not greater than unity. Hence, Q_k is either a line, or a half-line (ray), or a segment, which may degenerate to a point.

To each point $U=(b_1, b_2, \ldots, b_m, \lambda)\in Q_k$ there corresponds a set of points $X\in M$ so that $(C, X)=\lambda$. The linear-programming problem entails finding a point $X^*\in M$ maximizing (C, X). In the $(m+1)$-dimensional space, solving the problem is equivalent to finding a point $U^*\in Q_k$ with maximum possible $(m+1)$-th coordinate λ^*.

Two cases must be distinguished here:

1. There exists $\bar{\lambda}$ so that for any point $(b_1, b_2, \ldots, b_m, \lambda)\in Q_k$

$$\lambda\leqslant\bar{\lambda}.$$

2. The set Q_k contains points with arbitrarily large values of the $(m+1)$-th coordinate.

In the first case Q_k is either a segment or a ray with direction vector $-e_{m+1}$(a ray pointing along the negative Ou_{m+1}-axis). The problem in question is solvable and its optimal polyhedral set M^* (the set of all the solutions) corresponds to the upper (in the sense of the Ou_{m+1}-axis) end point of Q_k, i. e., the uppermost point of intersection of the cone K and the line Q.

If the set Q_k is a line or a ray with e_{m+1} as the direction vector, we have

the second case. This obviously indicates that the linear form of the problem is unbounded on the set of feasible programs.

It may occur that the line Q lies outside the cone K so that the intersection of Q and K is the empty set. In this case the problem has no feasible programs, i. e., the restraint system (4.21), (4.22) is inconsistent.

Thus, the solution of problem (4.20)–(4.22) in terms of the second geometrical interpretation entails finding the uppermost point of intersection of the line Q and the cone K. When Q and K have no common points, the problem is unsolvable since the restraints (4.21), (4.22) are inconsistent. If, however, the intersection of Q and K is a nonempty set having no uppermost point, the problem is unsolvable because the linear form (4.20) is unbounded on the polyhedral restraint set M.

In our geometrical interpretation of linear-programming problems, we have limited the discussion to maximization problems. All the foregoing applies, with obvious modifications, also to minimization problems. The reader will easily perform these modifications, if necessary.

EXERCISES TO CHAPTER 2

1. M_1 is an n-dimensional cube, i. e., a convex polyhedron, defined in the space of points $X=(x_1, x_2, \ldots, x_n)$ by the restraints

$$0 \leqslant x_j \leqslant a, \quad a>0, \qquad j=1, 2, \ldots, n.$$

Determine the number of ϱ-dimensional faces of M_1, $0 \leqslant \varrho \leqslant n$.

2. M_2 is an n-dimensional simplex, i. e., a convex polyhedron, defined in the space of points $X=(x_1, x_2, \ldots, x_{n+1})$ by the restraints

$$\sum_{j=1}^{n+1} x_j = 1;$$
$$x_j \geqslant 0, \qquad j=1, 2, \ldots, n+1.$$

Determine the number of ϱ-dimensional spaces of M_2, where $0 \leqslant \varrho \leqslant n$.

3. Solve the system of linear inequalities

$$\begin{aligned} x_1-2x_2+x_3 &\leqslant 1, \\ 2x_1+x_2-x_3 &\leqslant 2, \\ 3x_1+2x_2-2x_3 &\leqslant 3, \\ x_1-x_2-x_3 &\leqslant 1, \\ 2x_1+3x_2+2x_3 &\leqslant 2, \end{aligned}$$

giving its general solution.

Hint. Apply the representation theorem of polyhedral sets.

4. Let the faces Γ_0 and Γ of polyhedral sets M_0 and M be corresponding in the sense of 2-3. Prove that their dimensionalities $\varrho(\Gamma_0)$ and $\varrho(\Gamma)$ are related by

$$\varrho(\Gamma) = \varrho(\Gamma_0) + n - r,$$

where n is the dimensionality of the space and r is the rank of the restraint system defining the polyhedral set M.

5. Prove that in the n-dimensional vector space there exist $n+1$ vectors so that all their nonnegative linear combinations span the given space. Show that any l vectors with $l \leqslant n$ do not have this property.

6. Prove that a convex polyhedron may be defined as a closed bounded convex set containing a finite number of extremum points.

7. Show that any convex polyhedral set can be represented as a sum of some convex polyhedron and some convex polyhedral cone with apex at the origin.

8. Show that a convex polyhedral set with vertices is a convex hull* of its vertices and unbounded edges.

9. Prove that a convex polyhedral set M is a convex hull of its $n-r$- and $n-r+1$-dimensional faces. Here n is the dimensionality of the space of points X and r is the rank of the restraint system (1.1), (1.2)

* [For the definition of a convex hull, see p. 494.]

defining M. Show that in this representation only such $(n-r+1)$-dimensional faces need be retained which contain one $n-r$-dimensional face.

10. Prove that a convex polyhedral cone, which is not an intersection of several hyperplanes, coincides with the convex hull of its $n-r+1$-dimensional faces, where n and r are defined as in the preceding exercise.

11. Give an example of a continuous function defined on a convex polyhedral set to which Theorem 4.4 does not apply.

12. If the support problem X of (4.1)–(4.3) is degenerate, the number of edges of the polyhedral set M issuing from the vertex X may be either less, or more than $n-t$. Construct suitable examples.

13. Show that an arbitrary linear transformation

$$Y = AX,$$

where A is a $n_1 \times n_2$ matrix, transforms any convex polyhedral set of the n_2-dimensional space of points X into a convex polyhedral set of the n_1-dimensional space of points Y.

Chapter 3

DUALITY

This chapter deals with the principles of one of the basic concepts of linear programming — duality.

Associated with every linear-programming problem is a corresponding (linear-programming) problem, generally called the dual or the conjugate problem. Duality establishes a close relationship between the two problems constituting a unique dual pair. Simultaneous analysis of dual pairs proves to be useful both in developing numerical methods of linear programming and in qualitative investigations in linear programming and related mathematical subjects.

We present the theory in the following sequence. In § 1 a general statement of the problem is given and some elementary dual relationships are established. In § 2 certain properties of convex polyhedral cones, essential in the proof of duality theorems, are derived. In proving the theorems presented in § 3 for problems with homogeneous restraints we use the lucid geometrical interpretation as opposed to the conventional technique (see, e. g. , /25/). In § 4 duality theorems are extended to linear-programming problems with mixed restraints. In § 5 we deal with the relationship between the solutions of the dual problem and the so-called decision multipliers. Optimality criteria of programs are formulated in terms of decision multipliers. In this section we also give two significant interpretations of decision multipliers. According to the first interpretation decision multipliers can be considered as analogous to Lagrange multipliers, and according to the second, decision multipliers are used to assess the influence of the right-hand sides of the restraints on the maximum attainable value of the associated linear form.

In the last section (§ 6) duality theorems are used to establish certain sufficiency conditions for the unique solvability of the linear-programming problem and to prove certain propositions of the theory of linear inequalities.

§ 1. Statement of the problem

1-1. Consider a general linear-programming problem in canonical form. Maximize the linear form

$$L(X)=\sum_{j=1}^{n}c_jx_j \tag{1.1}$$

subject to the conditions

$$\sum_{j=1}^{n}a_{ij}x_j=b_i, \qquad i=1,2,\ldots,m; \tag{1.2}$$

$$x_j \geqslant 0, \qquad j=1, 2, \ldots, n. \tag{1.3}$$

In conjunction with problem (1.1)—(1.3) we introduce another linear-programming problem.

Minimize the linear form

$$\tilde{L}(Y)=\sum_{i=1}^{m} b_i y_i \tag{1.4}$$

subject to the conditions

$$\sum_{i=1}^{m} a_{ij} y_i > c_j, \qquad j=1, 2, \ldots, n. \tag{1.5}$$

Problem (1.4)–(1.5) is generally called dual or conjugate with respect to problem (1.1)–(1.3). Problem (1.1)–(1.3) is termed the primal problem.

We rewrite the restraints of problems (1.1)—(1.3) and (1.4)–(1.5) in matrix form. Let

$$A=\| a_{ij} \| = (A_1, A_2, \ldots, A_n)$$

be the restraint matrix of problem (1.1)–(1.3) comprising the restraint vectors A_j of this problem. Let, as before,

$$B=(b_1, b_2, \ldots, b_m)^T, \quad C=(c_1, c_2, \ldots, c_n).$$

With these notations, the above problems take the following form:

Primal problem. Determine an n-dimensional vector $X=(x_1, x_2, \ldots, x_n)^T$ maximizing

$$L(X)=(C, X) \tag{1.1'}$$

subject to the conditions

$$AX=B, \tag{1.2'}$$

$$X \geqslant 0. \tag{1.3'}$$

Dual problem. Determine an m-dimensional vector $Y=(y_1, y_2, \ldots, y_m)^T$ minimizing

$$\tilde{L}(Y)=(B, Y) \tag{1.4'}$$

subject to the conditions

$$A^T Y \geqslant C. \tag{1.5'}$$

The dual problem is, thus, obtained from the primal problem when:

(a) the vectors B and C are interchanged;

(b) the matrix A is transposed;

(c) the equalities of the restraints (1.2') are replaced by inequalities;

(d) the restraints (1.3') are eliminated;

(e) maximization is replaced by minimization.

We will now illustrate the above by a numerical example.

Consider a linear-programming problem which entails maximization of the linear form

$$L(X)=x_1+3x_2+2x_3-3x_5-x_6 \tag{1.6}$$

subject to the conditions

$$\left.\begin{aligned} 2x_1+2x_2+x_3+\ x_4+2x_5+\ x_6&=1, \\ 4x_1+3x_2-x_3-2x_4-\ x_5+2x_6&=1. \end{aligned}\right\} \tag{1.7}$$

$$x_j \geqslant 0, \quad j=1, 2, \ldots, 6. \tag{1.8}$$

Here

$$A=\begin{pmatrix}2&2&1&1&2&1\\4&3&-1&-2&-1&2\end{pmatrix};\quad B=(1,1)^T;\quad C=(1,3,2,0,-3,-1).$$

According to the general rules, the dual problem of problem (1.6)–(1.8) is formulated as follows. Minimize the linear form

$$\tilde{L}(Y)=y_1+y_2 \tag{1.9}$$

subject to the conditions

$$\left.\begin{aligned}2y_1+4y_2&\geqslant 1,\\2y_1+3y_2&\geqslant 3,\\y_1-y_2&\geqslant 2,\\y_1-2y_2&\geqslant 0,\\2y_1-y_2&\geqslant -3\\y_1+2y_2&\geqslant -1.\end{aligned}\right\} \tag{1.10}$$

1-2. We now give the geometrical interpretation of the dual problem. We first recall the geometrical meaning of the problem (1.1)–(1.3) according to the second geometrical interpretation (see Chapter 2, 5-3). The restraints of problem (1.1)–(1.3) define a linear transformation of the n-dimensional space of points $X=(x_1, x_2, \ldots, x_n)$ onto the $(m+1)$-dimensional space of points $U=(u_1, u_2, \ldots, u_{m+1})$ so that

$$u_i=\sum_{j=1}^{n} a_{ij}x_j, \qquad i=1, 2, \ldots, m;$$

$$u_{m+1}=\sum_{j=1}^{n} c_j x_j.$$

The image of the positive orthant of the X-space is a convex polyhedral cone K, with apex at the origin, generated by the augmented restraint vectors $\bar{A}_1, \bar{A}_2, \ldots, \bar{A}_n$. We remind the reader that

$$\bar{A}_j=(a_{1j}, a_{2j}, \ldots, a_{nj}, c_j)^T.$$

The image of the set consisting of all the solutions of (1.2) is a line Q passing through the point $\bar{B}=(b_1, b_2, \ldots, b_m, 0$ parallel to the Ou_{m+1}-axis. The equation of the line Q is

$$U=\bar{B}+\lambda e_{m+1}, \qquad -\infty<\lambda<\infty, \tag{1.11}$$

where

$$e_{m+1}=(\underbrace{0, 0, \ldots, 0,}_{m} 1).$$

Problem (1.1)–(1.3) entails finding the "upper" points of intersection of the line Q and the cone K.

Consider the set of all the hyperplanes in the $(m+1)$-dimensional space of points U passing through the origin. The equation of any of these hyperplanes is

$$\sum_{k=1}^{m+1} \lambda_k u_k=0. \tag{1.12}$$

The hyperplane (1.12) and its direction vectors $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m, \lambda_{m+1})$, specified to an arbitrary (nonzero) factor, are mutually orthogonal. For our purposes it is convenient to impose on the direction vector Λ the additional condition

$$\lambda_{m+1}=-1. \tag{1.13}$$

This condition eliminates hyperplanes parallel to the Ou_{m+1}-axis. From condition (1.13), there is a one-to-one correspondence between the hyperplanes passing through the origin and not containing the Ou_{m+1}-axis and their direction vectors.

Let $Y=(y_1, y_2, \ldots, y_m)$ be a feasible program of problem (1.4)–(1.5), i. e., a vector satisfying conditions (1.5). Consider the hyperplane Π_Y defined by the equation

$$\sum_{i=1}^{m} y_i u_i - u_{m+1} = 0. \tag{1.14}$$

We shall show that the cone K is located in one of the half-spaces generated by Π_Y.

According to (1.5), which the vector Y satisfies, insertion of any of the $\bar{A}_j$ ($j=1, 2, \ldots, n$) in the left-hand side of (1.14) gives a nonnegative number. Therefore, the position of the cone K generated by the vectors $\bar{A}_j$, $j=1, 2, \ldots, n$, relative to the hyperplane Π_Y is indicated by the vector $-e_{m+1}=(0, 0, \ldots, 0, -1)$, corresponding to the negative direction of the Ou_{m+1}-axis. The cone K is thus below the hyperplane Π_Y (in the sense of the Ou_{m+1}-axis).

Consider now an arbitrary hyperplane passing through the origin and not containing the Ou_{m+1}-axis. Let its equation be given by (1.14).

If the cone K is below the hyperplane (1.14), the vector $Y=(y_1, \ldots, y_m)$ defining this hyperplane satisfies the conditions (1.15) and, consequently, is a feasible program of the dual problem (1.4)–(1.5). To verify this proposition, it is sufficient to insert in the left-hand side of (1.14) the coordinates of the augmented restraint vector

$$\bar{A}_j (j=1, 2, \ldots, n),$$

which, by assumption, is above the hyperplane in question.

Thus, geometrically, the set of feasible programs of the dual problem coincides with the set of hyperplanes passing through the origin and situated above the cone K. Moreover, there is a one-to-one correspondence, defined by equation (1.14), between the programs Y of the dual problem and the hyperplanes Π_Y of the set.

Let us determine the $(m+1)$-th coordinate, u_{m+1}^{Y}, of the point of intersection of the line Q and the hyperplane Π_Y. Applying (1.11) and (1.14), we obtain

$$u_{m+1}^{(Y)} = \sum_{i=1}^{m} b_i y_i = \tilde{L}(Y). \tag{1.15}$$

The relationship (1.15) shows that the value of the linear form of the dual problem of program Y equals the "distance" of the point of intersection of the line Q and the hyperplane Π_Y from the hyperplane $u_{m+1}=0$ (the word "distance" appears in quotation marks because u_{m+1}^{Y} may be either positive or negative).

We now give a geometrical interpretation of the dual problem. Geometrically, the dual problem (1.4)–(1.5) consists of finding a hyperplane passing through the origin and situated above the cone K which, moreover, intersects the line Q at the lowest point (relative to the Ou_{m+1}-axis).

1-3. For $m=2$ the above geometrical considerations can be easily visualized.

Consider problem (1.6)–(1.8) with two equality restraints (1.7). In this case the augmented restraint vectors have the form

$$\bar{A}_1=\begin{pmatrix}2\\4\\1\end{pmatrix}, \quad \bar{A}_2=\begin{pmatrix}2\\3\\3\end{pmatrix}, \quad \bar{A}_3=\begin{pmatrix}1\\-1\\2\end{pmatrix}, \quad \bar{A}_4=\begin{pmatrix}1\\-2\\0\end{pmatrix}, \quad \bar{A}_5=\begin{pmatrix}2\\-1\\-3\end{pmatrix}, \quad \bar{A}_6=\begin{pmatrix}1\\2\\-1\end{pmatrix}.$$

The cone K is generated by the vectors $\bar{A}_j$, $j=1, 2, 3, 4, 5, 6$ and is located in the three-dimensional space of points $U=(u_1, u_2, u_3)$. The edges of the cone K are the rays

$$U=\bar{A}_j\lambda, \quad \lambda\geqslant 0, \qquad j=1, 2, 3, 4, 5, 6.$$

The equation of the line Q, according to (1.11), is

$$\left.\begin{aligned} u_1 &= 1, \\ u_2 &= 1, \\ u_3 &= \lambda, \end{aligned}\right\} \tag{1.16}$$

where $-\infty<\lambda<\infty$.

Let the cone K be intercepted by a plane passing through the point (1, 0, 0) and perpendicular to the Ou_1-axis. The equation of this plane is $u_1=1$. The intersection of the plane $u_1=1$ and the cone K is a polygon $a_1a_2a_3a_4a_5a_6$. The vertices of the polygon $a_1a_2a_3a_4a_5a_6$ are the points of intersection of the plane $u_1=1$ with the corresponding edges $U=\bar{A}_j\lambda$, $\lambda\geqslant 0$, of the polyhedral cone K.

The polygon $a_1a_2a_3a_4a_5a_6$ is shown in Figure 3.1. In this figure the line Q contained in the $u_1=1$ plane is also shown.

Each plane passing through the origin and not containing the Ou_3-axis intersects the plane $u_1=1$ along a line not parallel to Q. These lines are known as the traces of the corresponding planes. Figure 3.1 shows four such traces:

$$SR; \quad S'R'; \quad S''R''; \quad S^*R^*.$$

We observe that any line in Figure 3.1 is the trace of a plane corresponding to a feasible program of the dual problem (1.9)–(1.10) if and only if the polygon $a_1a_2a_3a_4a_5a_6$ is below this line. In particular, the traces RS and R^*S^* correspond to feasible programs of the dual problem. The two other traces

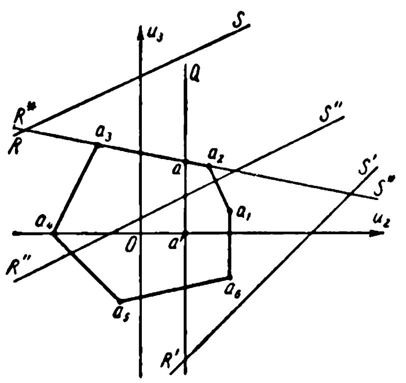


FIGURE 3.1

do not specify feasible programs of the dual problem: $R'S'$ passes below the polygon $a_1a_2a_3a_4a_5a_6$, and $R''S''$ "cuts" the polygon in two. The geometrical image of the solution of the primal problem is the point a, namely the upper point of intersection of the line Q and the polygon $a_1a_2a_3a_4a_5a_6$.

The solution procedure for a dual problem consists of finding the trace lying above the polygon $a_1a_2a_3a_4a_5a_6$ and intersecting the Q-axis at the lowest point. Geometrically, it is obvious that these requirements are fulfilled by R^*S^*, which is a supporting line of the polygon $a_1a_2a_3a_4a_5a_6$ at the point a. Hence, it follows that the linear forms of problems (1.6)–(1.8) and (1.9)–(1.10)

coincide with their respective optimal programs. The optimal value of either linear form is equal to the length of the segment aa' (see Figure 3.1).

In the following we shall see that the above properties of problems (1.6)–(1.8) and (1.9)–(1.10) are true for any linear-programming problem.

To clarify the geometrical meaning of the problem (1.9)–(1.10) we could, naturally, use the cone K, and not necessarily the intersection of the cone with the plane $u_1 = 1$ containing the Q-axis. The transition to the polygon $a_1a_2a_3a_4a_5a_6$ was effected only to make the discussion less abstract.

Observe that the plane $u_1 = 1$ could be replaced by any plane containing the line Q.

1-4. In Chapter 1, § 7 we gave an economic interpretation of the linear-programming problem and only considered the class of problems with inequality restraints and nonnegative variables. Problems of this class are called problems with *homogeneous restraints*. The economic interpretation of the dual problem (see 1-5) applies only to problems with homogeneous restraints. An arbitrary linear-programming problem with homogeneous restraints is formulated as follows:

Maximize the linear form

$$\sum_{j=1}^{n} c_j x_j \tag{1.17}$$

subject to the conditions

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \qquad i = 1, 2, \ldots, m; \tag{1.18}$$

$$x_j \geqslant 0, \qquad j = 1, 2, \ldots, n. \tag{1.19}$$

Problem (1.17)–(1.19) is easily reduced to canonical form. To this end it suffices to introduce additional nonnegative variables x_{n+i}, $i = 1, 2, \ldots, m$, and to rewrite the restraints of the problem (1.17)–(1.19) in the equivalent form

$$\sum_{j=1}^{n} a_{ij} x_j + x_{n+i} = b_i, \qquad i = 1, 2, \ldots, m; \tag{1.18'}$$

$$x_j \geqslant 0, \qquad j = 1, 2, \ldots, n+m. \tag{1.19'}$$

The equivalent problem has $n+m$ nonnegative variables and m equality restraints.

Applying the general rules we formulate the dual problem of (1.17), (1.18') and (1.19').

The augmented restraint vectors of the primal problem are

$$\bar{A}_j = \begin{cases} (a_{1j}, a_{2j}, \ldots, a_{mj}, c_j)^T, & \text{if } 1 \leqslant j \leqslant n, \\ (\underbrace{0, 0, \ldots, 0}_{j-n}, 1, 0, \ldots 0)^T, & \text{if } n+1 \leqslant j \leqslant n+m. \end{cases}$$

The dual problem is therefore formulated as follows.

Minimize the linear form

$$\sum_{i=1}^{m} b_i y_i \tag{1.20}$$

subject to the conditions

$$\sum_{i=1}^{m} a_{ij} y_i \geqslant c_j, \qquad j = 1, 2, \ldots, n; \tag{1.21}$$

$$y_i \geqslant 0, \qquad i = 1, 2, \ldots, m. \tag{1.22}$$

It must be emphasized that the dual problem (1.20)–(1.22) is a problem with homogeneous restraints.

Introducing new parameters $a'_{ij} = -a_{ij}$, $b'_i = -b_i$, $c'_j = -c_j$ we reduce the problem (1.20)–(1.22) to the maximization of the linear form

$$\sum_{i=1}^{m} b'_i y_i \tag{1.20'}$$

subject to the conditions

$$\sum_{i=1}^{m} a'_{ij} y_i \leqslant c'_j, \qquad j = 1, 2, \ldots, n; \tag{1.21'}$$

$$y_i \geqslant 0, \qquad i = 1, 2, \ldots, m. \tag{1.22'}$$

The ensuing problem is identical in form to the problem (1.17)–(1.19). Hence, its dual problem involves the minimization of the linear form

$$\sum_{j=1}^{n} c'_j x_j \tag{1.23}$$

subject to the conditions

$$\sum_{j=1}^{n} a'_{ij} x_j \geqslant b'_i, \qquad i = 1, 2, \ldots, m; \tag{1.24}$$

$$x_j \geqslant 0, \qquad j = 1, 2, \ldots, n. \tag{1.25}$$

If in problem (1.23)–(1.25) we revert to the original parameters a_{ij}, b_i, c_j we obtain the primal problem (1.17)–(1.19) which is thus the dual of the problem (1.20)–(1.22). Consequently, the problems (1.17)–(1.19) and (1.20)–(1.22) may, obviously, be considered as a *dual*, or *conjugate pair*. Each problem of this pair is thus dual with respect to the other. To single out one of the problems of a conjugate pair as the primal problem is thus, to a certain extent, arbitrary and purely formal.

In § 3 we shall see that the problems (1.1)–(1.3) and (1.4)–(1.5) also constitute a dual pair.

1-5. We now give an economic interpretation of the problem (1.20)–(1.22), dual with respect to the problem (1.17)–(1.19). First we recall the economic interpretation of the primal problem (1.17)–(1.19) given in Chapter 1, 7-2.

There are n modes for producing a certain homogeneous commodity, of which c_j elements are produced by the j-th production mode in unit time. Operating by the j-th production mode for unit time involves the consumption of the i-th production factor ($i = 1, 2, \ldots, m$) in the amount of a_{ij}.

Let the resources of the production factors be $b_1, b_2, \ldots, b_m$ units, respectively. Let x_j be the time scheduled for the j-th mode of production.

Problem (1.17)–(1.19) is the mathematical statement of the problem dealing with the determination of a scheduling program of the various production modes ensuring a maximum output of the homogeneous commodity with the given resources ($b_1, b_2, \ldots, b_m$).

Now, suppose that within the framework of the problem specified we have to evaluate each of the production factors. We shall consider here only an idealized self-contained model of production where the external relationships are strictly specified by the conditions of the problem.

The problem restraints (limited resources and available production modes) enable us to evaluate each of the factors. It should be kept in mind that this evaluation is characteristic for the industry in question and as such is relative. Similar production factors are different in significance in different industries and localities. In 5-5 we shall see that factor evaluation is a measure of usefulness of a particular factor for the *given* industry under *strictly specified* conditions.

Any change in production conditions, in particular a change in the resources of the various factors, makes it necessary to reassess these factors. The evaluation of the production factors is relative because these factors are assessed in units of value of the output commodity. The value of the commodity is determined by conditions which are not inherent in the production process.

Let the cost of a unit of the output commodity be taken as unity. The cost of a unit of the output commodity (the production cost) is introduced here as a basic concept proceeding from which we shall evaluate the various production factors. Let y_i $(i=1, 2, \ldots, m)$ be the cost of the i-th production factor.

We now study the j-th mode of production in terms of cost and profit.

Producing under this mode for a unit time, we incur an overall cost

$$z_j = \sum_{i=1}^{m} a_{ij} y_i,$$

whereas the cost of the output commodity is c_j. Given the correct values of the production factors, the overall expenditure cannot be less than the cost of the output commodity, since otherwise the goods would have been partly produced from "nothing". Hence, for any $j=1, 2, \ldots, n$,

$$z_j \geqslant c_j.$$

In other words, the cost vector $Y=(y_1, y_2, \ldots, y_m)$ should satisfy conditions (1.21).

Moreover, the costs $y_1, y_2, \ldots, y_m$ should obviously be taken as nonnegative numbers. The vector $Y=(y_1, y_2, \ldots, y_m)$, therefore, also satisfies condition (1.22).

The cost vector of the production factors is thus a feasible program of the dual problem (1.20)–(1.22).

However, conditions (1.21), (1.22) do not completely specify the cost vector Y. When interpreting the problem (1.17)–(1.19) economically, the parameters a_{ij} should naturally be taken as nonnegative numbers and, for any j, at least one of the a_{ij} should not vanish. Any vector with sufficiently large components is, therefore, a feasible program of the dual problem (1.20)–(1.22).

We must, therefore, find a condition which would eliminate unjustified exaggeration of the cost of the production factors. An obvious restriction of this kind is the following. The cost vector Y should be such that the overall value of the resources,

$$\sum_{i=1}^{m} b_i y_i,$$

at the disposal of the factory be as small as possible.

In what follows we shall see that if this requirement is not met with, the cost of the output commodity, for any scheduling program of the production modes, is invariably less than the value of all the resources. Conversely, if the foregoing condition is satisfied, there exist production programs ensuring equality of the cost of the output commodity to the resources available. This point also justifies the introduction of the last restraint.

The dual problem (1.20)–(1.22) is thus a mathematical statement of the problem of correct evaluation (costing) of all the production factors. The cost vector of the production factors solves the dual problem. In the following we shall sometimes call the feasible programs of the dual problem *tentative cost* vectors of the production factors.

The dual pair (1.17)–(1.19) and (1.20)–(1.22) is conveniently depicted in the following tableau (Table 3.1).

TABLE 3.1

		Productivity		
		$c_1 \quad c_2 \quad \dots \quad c_j \quad \dots \quad c_n$		
Cost vector	y_1	$a_{11} \quad a_{12} \quad \dots \quad a_{1j} \quad \dots \quad a_{1n}$	b_1	Resources
	y_2	$a_{21} \quad a_{22} \quad \dots \quad a_{2j} \quad \dots \quad a_{2n}$	b_2	
	$\vdots$	$\dots\dots\dots\dots\dots$	$\vdots$	
	y_i	$a_{i1} \quad a_{i2} \quad \dots \quad a_{ij} \quad \dots \quad a_{in}$	b_i	
	$\vdots$	$\dots\dots\dots\dots\dots$	$\vdots$	
	y_m	$a_{m1} \quad a_{m2} \quad \dots \quad a_{mj} \quad \dots \quad a_{mn}$	b_m	
		$x_1 \quad x_2 \quad \dots \quad x_j \quad \dots \quad x_n$		
		Production program		

1-6. To conclude this section we advance some simple, but nevertheless most useful, propositions relating to primal and dual programs. These propositions are repeatedly used in the following.

Lemma 1.1. If $X=(x_1, x_2, \dots, x_n)$ *and* $Y=(y_1, y_2, \dots, y_m)$ *are feasible programs of problems (1.1)–(1.3) and (1.4)–(1.5), respectively, then*

$$\sum_{j=1}^{n} c_j x_j \leqslant \sum_{i=1}^{m} b_i y_i. \tag{1.26}$$

Proof. By assumption,

$$\sum_{i=1}^{m} a_{ij} y_i \geqslant c_j, \qquad j=1, 2, \dots, n.$$

Hence,

$$\sum_{j=1}^{n} c_j x_j \leqslant \sum_{j=1}^{n} x_j \sum_{i=1}^{m} a_{ij} y_i = \sum_{i=1}^{m} y_i \sum_{j=1}^{n} a_{ij} x_j.$$

The vector X is a feasible program of problem (1.1)–(1.3); therefore

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \text{ for } i=1, 2, \dots, m.$$

Applying these equalities, we transform the right-hand side of the preceding relationships and obtain

$$\sum_{j=1}^{n} c_j x_j \leqslant \sum_{i=1}^{m} b_i y_i.$$

This completes the proof.

From Lemma 1.1 it follows, in particular, that if we take a feasible program of the dual problem (which is not its solution) as the cost vector we shall not be able to equate the cost of the output commodity and the resources available.

Indeed, if Y is a nonoptimal production program of the dual problem, and Y^* is the solution of this problem, then for any primal program X

$$\sum_{j=1}^{n} c_j x_j \leqslant \sum_{i=1}^{m} b_i y_i^* < \sum_{i=1}^{m} b_i y_i.$$

Lemma 1.2. If for some programs $X^*=(x_1^*, x_2^*, \dots, x_n^*)$ *and* $Y^*=(y_1^*, y_2^*, \dots, y_m^*)$

of problems (1.1)-(1.3) and (1.4)-(1.5), respectively,

$$\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} b_i y_i^*, \tag{1.27}$$

the vectors X^*, Y^* are solutions of the corresponding problems.

Proof. According to Lemma 1.1, for any program $X=(x_1, x_2, \ldots, x_n)$ of problem (1.1)-(1.3),

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{i=1}^{m} b_i y_i^*.$$

Hence, applying the conditions of Lemma 1.2, we obtain

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{j=1}^{n} c_j x_j^*. \tag{1.28}$$

Inequality (1.28), which holds for any feasible program X of problem (1.1)-(1.3), shows that X^* is indeed the optimal program.

The optimality of program Y^* of the dual problem (1.4)-(1.5) is established analogously. This completes the proof.

Lemma 1.2 can be interpreted economically thus. If for some feasible scheduling program of production modes X^* and some tentative cost vector of production factors Y^* the cost of the output commodity is equal to the overall value of the resources available, X^* and Y^* are, respectively, the optimal production program and the optimal cost vector.

Lemma 1.2 establishes sufficiency of conditions (1.27) for optimality of programs X^* and Y^*. Below we shall see that equality (1.27) is also a necessary condition for the optimality of programs X^* and Y^*.

Lemma 1.3. *If the linear form (1.4) of the dual problem (1.4)-(1.5) is unbounded from below on the set of its feasible programs, the primal problem (1.1)-(1.3) has no feasible programs.*

Proof. By assumption, there exists a set of feasible programs $\{Y_k\}$ of the dual problem (1.4)-(1.5) such that

$$\lim_{k \to \infty} (B, Y_k) = -\infty. \tag{1.29}$$

If we assume that the primal problem (1.1)-(1.3) has a feasible program X, then according to Lemma 1.1

$$(C, X) \leq (B, Y_k)$$

for any natural k. Passing to the limit as $k \to \infty$ and applying (1.29), we obtain

$$(C, X) = -\infty. \tag{1.30}$$

If all the components of the vector X are finite, equality (1.30) is meaningless.

Hence the assumption that there exists at least one feasible program of problem (1.1)-(1.3) is false. This completes the proof.

We now elucidate the geometrical meaning of Lemma 1.3 in the $(m+1)$-dimensional space of points

$$U = (u_1, u_2, \ldots, u_{m+1}).$$

If there exists a set of hyperplanes passing through the origin, situated above the cone K, and intersecting the Q-axis at points whose $(m+1)$-th coordinate tends to $-\infty$, the line Q and the cone K are disjoint.

Lemma 1.3 proves useful when it is necessary to establish the solvability of a linear-programming problem.

All three lemmas given above were formulated for a canonical linear-programming problem. The same propositions hold, obviously, also for problems with homogeneous restraints since in constructing the dual problem of problem (1.17)-(1.19) we reduced the primal problem to canonical form. Furthermore, this property has been used implicitly in our economic illustration of the propositions.

§ 2. On some properties of convex polyhedral cones

2-1. In the proof of duality theorems in the next section certain properties of convex polyhedral cones are used. These properties will be established in the present section. In the following, when speaking of points or vectors we shall refer to elements of an n-dimensional vector space, n being arbitrary.

In Chapter 2 we advanced two equivalent definitions of a convex polyhedral cone in an n-dimensional space (see 3-5).

Here it is more convenient to operate with the second definition, which is as follows.

The set K_{P_0} comprising the points

$$P=\sum_{i=1}^{N}\beta_i R_i+P_0,$$

where $\beta_i \geqslant 0$, $i=1, 2, \ldots, N$ and $R_1, R_2, \ldots, R_N$, P_0 are some points (vectors) of the n-dimensional space, is called a convex polyhedral cone with apex at P_0 spanned (generated) by the vectors $R_1, R_2, \ldots, R_N$.

We give one elementary property of convex cones (for the corresponding definition, see Appendix, 3-2).

Lemma 2.1. Let Π *be the support hyperplane of the cone* T *at the point* P. *Then this hyperplane contains any point* P' *of the form*

$$P'=P_0+\mu(P-P_0), \tag{2.1}$$

where P_0 *is the apex of* T, *and* $\mu \geqslant 0$.

Proof. Let the equation of the hyperplane Π be

$$(\Lambda, X)=c. \tag{2.2}$$

To be specific, assume that μ in (2.1) is larger than 1 (the case $\mu<1$ can be dealt with analogously).

If P'' is any point of the cone T which may be represented by (2.1) for $\mu<1$, there exists a number S, $0<S<1$, such that

$$P=SP'+(1-S)P''.$$

By assumption

$$(\Lambda, P)=c.$$

Hence

$$(\Lambda, SP'+(1-S)P'')=c,$$

or, after obvious manipulations,

$$S[(\Lambda, P')-c]+(1-S)[(\Lambda, P'')-c]=0. \tag{2.3}$$

Since Π is a support hyperplane of the cone $\mathbf{T}$, each of the elements in brackets in (2.3) is nonpositive (or nonnegative). Therefore, for equality (2.3) to hold for $0<S<1$ it is necessary that both brackets vanish, i. e.,

$$(\Lambda, P') = (\Lambda, P'') = c.$$

Thus, any point P' located on a ray issuing from P_0 in the direction of P belongs to the support hyperplane Π. This completes the proof.

Applying Lemma 2.1 for $\mu=0$, we obtain

Corollary 2.1. Any support hyperplane of a convex cone contains its apex.

Lemma 2.1 and its corollary apply to any convex cone. In particular, they hold for convex polyhedral cones.

2-2. The propositions dealt with below hold only for convex polyhedral cones.

Consider a convex polyhedral cone K spanned by the vectors $R_1, R_2, \ldots, R_N$ with apex at the origin. Let the support hyperplane of the cone K at the point $Q \in K$ be denoted by Π_Q.

Lemma 2.2. For any boundary point P of the cone K there exists $\varepsilon>0$ such that any support hyperplane Π_Q, where $|P-Q|<\varepsilon$, contains the point P.*

Proof. Let us assume the contrary, i. e., there exists a sequence $\{Q_\alpha\}$ of boundary points of the cone K such that

$$\lim_{\alpha \to \infty} |P - Q_\alpha| = 0,$$

but none of the hyperplanes Π_{Q_α} contains the point P.

Let

$$Q_\alpha = \sum_{k=1}^{s_\alpha} \beta_{i_k}^{(\alpha)} R_{i_{k,\alpha}},$$

where $\beta_{i_k}^{(\alpha)}>0$, $k=1, 2, \ldots, s_\alpha$. There is an infinite number of points Q_α. The number of different systems comprising the vectors $R_1, R_2, \ldots, R_N$ is a priori finite. Therefore, from the sequence $\{Q_\alpha\}$ we can find an infinite subsequence $\{Q_{\alpha_t}\} = \{\bar{Q}_t\}$ such that

$$\bar{Q}_t = \sum_{k=1}^{s} \beta_k^{(t)} R_{i_k},$$

where the system of vectors $R_{i_1}, R_{i_2}, \ldots, R_{i_s}$ is the same for all the points $Q_{\alpha_t} = \bar{Q}_t$, $\beta_k^{(t)}>0$, $k=1, 2, \ldots, s$, $t=1, 2\ldots$ We have thus formed a sequence $\{\bar{Q}_t\}$ of boundary points of the cone K with the following properties:

(a) for any t,

$$\bar{Q}_t = \sum_{k=1}^{s} \beta_k^{(t)} R_{i_k}, \tag{2.4}$$

where the system of vectors $R_{i_1}, R_{i_2}, \ldots, R_{i_s}$ is independent of the point $\bar{Q}_t$, and

$$\beta_k^{(t)}>0 \text{ for } k=1, 2, \ldots, s; \qquad t=1, 2, \ldots;$$

(b) $\lim_{t \to \infty} |\bar{Q}_t - P| = 0;$

(c) $P \notin \Pi_{\bar{Q}_t}$ for any $t=1, 2, \ldots$

We shall show that such a sequence cannot exist.

* We remind the reader that the symbol $|A|$ denotes the length of vector A: $|A| = \sqrt{(A, A)}$.

2154

Let $\bar{K}$ denote a polyhedral cone spanned by the vectors $R_{i_1}, R_{i_2}, \ldots, R_{i_s}$ with apex at the origin. According to (a),

$$\bar{Q}_t \in \bar{K}, \qquad t = 1, 2, \ldots$$

Condition (b) indicates that P is the cluster point of the sequence $\{\bar{Q}_t\}$. Since the convex polyhedral cone $\bar{K}$ is closed (see Chapter 2, Lemma 3.3), $P \in \bar{K}$. Hence,

$$P = \sum_{k=1}^{s} \beta_k R_{i_k}, \tag{2.5}$$

where $\beta_k \geqslant 0$, $k = 1, 2, \ldots, s$.

Consider an arbitrary point $\bar{Q}_t$ of the sequence $\{\bar{Q}_t\}$. Let the equation of the support hyperplane $\Pi_{\bar{Q}_t}$ be

$$(\Lambda_t, X) = 0. \tag{2.6}$$

To obtain equation (2.6) we applied Corollary 2.1 according to which the hyperplane $\Pi_{\bar{Q}_t}$ contains the point O, which is the apex of the cone $\bar{K}$.

Allowing for representation (2.4) of the point $\bar{Q}_t \in \Pi_{\bar{Q}_t}$, we have

$$(\Lambda_t, \bar{Q}_t) = \sum_{k=1}^{s} \beta_k^{(t)} (\Lambda_t, R_{i_k}) = 0. \tag{2.7}$$

Since $(\Lambda_t, R_{i_k}) \leqslant 0$ ($\Pi_{\bar{Q}_t}$ is a support hyperplane of the cone K), and $\beta_k > 0$, equality (2.7) applies only if

$$(\Lambda_t, R_{i_k}) = 0, \qquad k = 1, 2, \ldots, s. \tag{2.8}$$

Relationships (2.5) and (2.8) yield

$$(\Lambda_t, P) = 0,$$

which indicates that $P \in \Pi_{\bar{Q}_t}$.

None of the points of the sequence $\{\bar{Q}_t\}$ satisfying (a) and (b) can satisfy (c). This contradiction establishes the proposition of Lemma 2.2.

2-3. Let T_P be a polyhedral cone with apex at the point P spanned by nonzero vectors $T_1, T_2, \ldots, T_{N_1}$.

We shall assume the polyhedral cone T_P to satisfy the following condition:

For any $y_i \geqslant 0$,

$$\sum_{i=1}^{N_1} y_i T_i = 0$$

only if

$$y_1 = y_2 = \ldots = y_{N_1} = 0.$$

It can be shown (see Exercise 3) that this condition is equivalent to the requirement that the apex P of the cone T_P be its extreme point.

Lemma 2.3. If the intersection of convex polyhedral cones K *and* T_P *contains only one point* P, *there exists a hyperplane* Π *with equation*

$$(\Lambda, X) = 0,$$

which is the support hyperplane of K *at the point* P ($(\Lambda, X) \leqslant 0$ *if* $X \in K$; $(\Lambda, P) = 0$) *such that*

$$(\Lambda, Q) > 0 \tag{2.9}$$

for all the points Q *of the cone* T_P, *except its apex* P.

Proof. Consider the set $T_P^{(\delta)}$ $(0<\delta<1)$ comprising the points

$$Q=\sum_{k=1}^{N_1}\beta_k T_k+P\in T_P,$$

such that

$$\delta\leqslant\sum_{k=1}^{N_1}\beta_k\leqslant 1.$$

It can easily be verified that $T_P^{(\delta)}$ is a closed bounded convex set.

The proof is left to the reader (see Exercise 4).

The point $P\notin T_P^{(\delta)}$. Indeed, if $0<\delta\leqslant\sum_{k=1}^{N_1}\beta_k$, then according to the property, given above, of the cone T_P

$$\sum_{k=1}^{N_1}\beta_k T_k\neq 0,$$

so that any point P' of the set $T_P^{(\delta)}$ has the form

$$P'=P+P'',$$

where $P''\neq 0$, i.e., $P'\neq P$.

The sets K and $T_P^{(\delta)}$ are thus disjoint. Let P_δ be the point of the cone K least removed from the set $T_P^{(\delta)}$. The existence of such a point follows from the fact that the set K is closed and the set $T_P^{(\delta)}$ is bounded (see Appendix, 3-3). P_δ is obviously a boundary point of the cone K. Let Π_δ be the support hyperplane of K at the point P_δ such that the set $T_P^{(\delta)}$ lies in the half-space generated by Π_δ and does not contain K. (Such a point indeed exists; see Appendix, Corollary 3.2.)

We shall show that

$$\lim_{\delta\to 0}|P_\delta-P|=0. \tag{2.10}$$

Indeed, otherwise we could form a sequence $\{P_{\delta_k}\}$ converging to the point $\bar{P}\neq P$ (we use here the boundedness of the set of points P_δ, $0<\delta<1$).

Obviously, $P'_\delta=P+\delta T_1\in T_P^{(\delta)}$, so that

$$|P'_\delta-P|=\delta|T_1|.$$

Hence, by definition of P_δ, there exists a point $\bar{P}_\delta\in T_P^{(\delta)}$ such that

$$|\bar{P}_\delta-P_\delta|\leqslant\delta|T_1|.$$

Passing in this equality to the limit as $\delta_k\to 0$, $k=1, 2, \ldots$, we have

$$\lim_{k\to\infty}\bar{P}_{\delta_k}=\lim_{k\to\infty}P_{\delta_k}=\bar{P}.$$

It follows from this relationship and from the fact that the sets K and T_P are closed that the point $\bar{P}$ belongs to the intersection of the cones K and T_P (by assumption, $\bar{P}_\delta\in T_P$, $P_\delta\in K$). But the intersection of K and T_P contains only one point P. The ensuing contradiction proves the validity of (2.10).

Let us now take $\delta=\delta^*>0$ so small that

$$|P-P_{\delta^*}|<\varepsilon,$$

where ε is the quantity entering the conditions of Lemma 2.2. According to this lemma, Π_{δ^*} is the support hyperplane of the cone K at the point P.

Assume the hyperplane Π_{δ^*} to have an equation of the form

$$(\Lambda, X)=0.$$

From the definition of the hyperplane Π_{δ^*} we then have

$$(\Lambda, X) \leqslant 0 \text{ for } X \in K, \tag{2.11}$$
$$(\Lambda, P) = 0, \tag{2.12}$$
$$(\Lambda, X) > 0 \text{ for } X \in T_P^{(\delta^*)}. \tag{2.13}$$

Let Q be an arbitrary point of the cone T_P, other than P.
Then, obviously, there exists a number $\mu > 0$ such that

$$Q_\mu = P + \mu(Q - P) \in T_P^{(\delta^*)}.$$

Hence, applying (2.12) and (2.1 3) we have

$$(\Lambda, Q_\mu) = (\Lambda, (1-\mu)P) + (\Lambda, \mu Q) = \mu(\Lambda, Q) > 0.$$

Thus,

$$(\Lambda, Q) > 0 \tag{2.14}$$

for all $Q \in T_P$ other than P. Relationships (2.11), (2.12), and (2.14) indicate that the hyperplane Π_{δ^*} satisfies all the requirements of Lemma 2.3. This completes the proof.

Lemma 2.3 is often used in proving the duality theorems in the next section. In the proof of this lemma it is assumed that the convex cones K and T_P are polyhedral, and P is the point (i. e. , the extreme point) of the cone T_P. Many examples can be given in which departure from these assumptions as regards the cone K and the point P render the proposition of the lemma, generally speaking, invalid. As regards the cone T_P, it need not be polyhedral for Lemma 2.3 to hold.

The proof of these propositions is left to the reader (see Exercises 5 and 6).

§ 3. Duality theorems

3-1. We have shown in § 1 that either linear-programming problem with homogeneous restraints (1.17)–(1.19) and (1.20)–(1.22) is dual with respect to the other. These two problems thus constitute a dual (conjugate) pair of linear-programming problems. Conjugate linear-programming problems have certain interesting and important properties which place them in one class rather significant in applications. Some elementary properties of dual problems were considered in the end of § 1.

In this section we formulate and prove two principal duality theorems and derive from them some useful corollaries. The proof of the two theorems is based on properties of convex polyhedral cones which have been discussed in the preceding section. The duality theorems are proved here for canonical linear-programming problems with homogeneous restraints.

The general case is considered in the next section, where the concept of duality is extended to linear-programming problems given in arbitrary form.

3-2. Consider a general linear-programming problem in canonical form (problem (1.1)–(1.3)). Problem (1.1)–(1.3) and its dual problem (1.4)–(1.5) have the following important property:

Theorem 3.1. *If problem (1.1)–(1.3) has an optimal program, problem (1.4)–(1.5) is also solvable. Moreover, for any optimal programs* $X = (x_1, x_2, \ldots, x_n)$ *and* $Y = (y_1, y_2, \ldots, y_m)$ *of problems (1.1)–(1.3) and (1.4)–(1.5),*

respectively, the equality

$$\sum_{j=1}^{n} c_j x_j = \sum_{i=1}^{m} b_i y_i \tag{3.1}$$

holds.

Proof. 1. The proof is based on the following geometrical considerations (see 1-2). Consider a convex polyhedral cone K spanned by the augmented restraint vectors of the primal problem. The geometrical image of the optimal program of problem (1.1)-(1.3) is the "highest" point of intersection of line Q and cone K (the point $\bar{B}$). Applying Lemma 2.3 to the cone K and to the ray issuing upwards along the line $\bar{B}$ from the point Q, we construct the support hyperplane Π^* of K at the point $\bar{B}$ such that the vector e_{m+1} drawn from Π^* is above $\bar{B}$. This hyperplane corresponds to the optimal program of the dual problem.

2. We now proceed with a rigorous proof of the theorem. As usual, let $\bar{A}_j$ be the j-th augmented restraint vector of problem (1.1)-(1.3):

$$\bar{A}_j = (a_{1j}, a_{2j}, \ldots, a_{mj}, c_j)^T.$$

Let further

$$\bar{B} = (b_1, b_2, \ldots, b_m, \Delta)^T,$$

where $B = (b_1, b_2, \ldots, b_m)^T$ is the constraint vector of problem (1.1)-(1.3), and Δ is the maximum of the linear form (according to the conditions of the theorem, problem (1.1)-(1.3) is solvable).

Let $X^* = (x_1^*, x_2^*, \ldots, x_n^*)$ be the optimal program of problem (1.1)-(1.3). In this case, according to the definition of the vectors $\bar{A}_j$, $j = 1, 2, \ldots, n$, and $\bar{B}$,

$$\bar{B} = \sum_{j=1}^{n} \bar{A}_j x_j^*. \tag{3.2}$$

Let K denote the convex polyhedral cone with its apex at the origin spanned by the vectors $\bar{A}_j$, $j = 1, 2, \ldots, n$. Relationship (3.2) indicates that $\bar{B} \in K$. Consider the ray S issuing from the point $\bar{B}$ and parallel to the direction vector $e_{m+1} = (0, 0, \ldots, 0, 1)^T$. It is obvious that any point of the ray S, other than its end point $\bar{B}$, lies outside the cone K. Indeed, if for some $\mu > 0$ the point $\bar{B} + \mu e_{m+1}$ belongs to the cone K, there exist nonnegative numbers $\bar{x}_j$, $j = 1, 2, \ldots, n$, such that

$$\bar{B} + \mu e_{m+1} = \sum_{j=1}^{n} \bar{A}_j \bar{x}_j. \tag{3.3}$$

It follows from the vector equality (3.3) that $\bar{X} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ is a feasible program of problem (1.1)-(1.3) and that

$$\sum_{j=1}^{n} c_j \bar{x}_j = \Delta + \lambda > \Delta. \tag{3.4}$$

Relationship (3.4) contradicts the assumption that Δ is the maximum of the linear form (1.1) subject to the conditions (1.2), (1.3). Consequently, equality (3.3) is invalid, i. e. , for $\lambda > 0$

$$\bar{B} + \lambda e_{m+1} \notin K.$$

All the points of the ray S, except for the end point $\bar{B}$, thus lie outside K.

The ray S may be considered as a convex polyhedral cone spanned by the single vector e_{m+1} with its apex at the point $\bar{B}$. The apex $\bar{B}$ of the cone S is obviously its point.

Lemma 2.3 thus applies to cone K and to ray S. In this case $N_1=1$; $P=\bar{B}$; $T_1=e_{m+1}$; $T_P=S$. According to Lemma 2.3, there exists a hyperplane Π^* with equation

$$(\Lambda, U)=0,$$

such that

$$(\Lambda, \bar{A}_j)\leqslant 0, \quad j=1, 2, \ldots, n; \tag{3.5}$$

$$(\Lambda, \bar{B})=0; \tag{3.6}$$

$$(\Lambda, \bar{B}+e_{m+1})=(\Lambda, e_{m+1})>0. \tag{3.7}$$

Condition (3.5) follows from the fact that Π^* is a support hyperplane of the cone K. Equality (3.6) indicates that hyperplane Π^* contains the point $\bar{B}$, the apex of the cone S. Relationship (3.7) follows from condition (2.9) of Lemma 2.3.

3. We may now easily establish the optimal program of the dual program (1.4)-(1.5). By virtue of (3.7), the $(m+1)$-th component of the vector $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m, \lambda_{m+1})$ is positive:

$$(\Lambda, e_{m+1})=\lambda_{m+1}>0.$$

Applying this fact, we define an m-dimensional vector $Y^*=(y_1^*, y_2^*, \ldots, y_m^*)$, taking

$$y_i^*=-\frac{\lambda_i}{\lambda_{m+1}}, \qquad i=1, 2, \ldots, m.$$

Rewriting condition (3.5) in an obviously equivalent form, we have

$$(\Lambda, \bar{A}_j)=\sum_{i=1}^{m}\lambda_i a_{ij}+\lambda_{m+1}c_j\leqslant 0,$$

or

$$\sum_{i=1}^{m}\left(-\frac{\lambda_i}{\lambda_{m+1}}\right)a_{ij}=\sum_{i=1}^{m}y_i^*\, a_{ij}\geqslant c_j, \qquad j=1, 2, \ldots, n.$$

The vector Y^* thus satisfies conditions (1.5) and as such is a feasible program of the dual problem (1.4)-(1.5).

We now prove the optimality of program Y^*. Expressing the scalar product in the left-hand side of equality (3.6) in terms of the coordinates of the vectors Λ and $\bar{B}$, we obtain

$$\sum_{i=1}^{m}\lambda_i b_i+\lambda_{m+1}\Delta=0,$$

or

$$\sum_{i=1}^{m}y_i^*\, b_i=\Delta.$$

According to the $(m+1)$-the component of the vector equality (3.2)

$$\sum_{j=1}^{n}x_j^*\, c_j=\Delta.$$

Hence,

$$\sum_{j=1}^{n}x_j^*\, c_j=\sum_{i=1}^{m}y_i^*\, b_i=\Delta. \tag{3.8}$$

According to Lemma 1.2 equality (3.8) proves the optimality of program Y^* of problem (1.4)-(1.5).

The dual problem is thus solvable and its optimal program Y^* satisfies equality (3.8).

If, now, X and Y are two solutions of problems (1.1)-(1.3) and (1.4)-(1.5),

respectively, we have

$$\sum_{j=1}^{n} x_j c_j = \sum_{j=1}^{n} x_j^* c_j = \Delta,$$

$$\sum_{i=1}^{m} y_i b_i = \sum_{i=1}^{m} y_i^* b_i = \Delta,$$

so that

$$\sum_{j=1}^{n} x_j c_j = \sum_{i=1}^{m} y_i b_i.$$

This completes the proof.

From this theorem follows the so-called **first duality theorem** applicable to problems with homogeneous restraints.

Theorem 3.2. (First duality theorem.) If either problem of a dual pair (1.17)-(1.19) and (1.20)-(1.22) is solvable, the other problem is also solvable. Any two optimal programs X and Y of these problems satisfy equality (3.1).

Proof. Consider an arbitrary problem of a given dual pair. In § 1 we showed that if it is reduced to canonical form and then its dual problem is formulated, we obtain the second problem of the dual pair. Therefore, to prove Theorem 3.2 it suffices to refer to Theorem 3.1.

We observe that the first duality theorem can also be proved without applying the comparatively complicated Lemma 2.3 (see Exercise 9).

3-3. We give some corollaries of the first duality theorem.

Corollary 3.1. One of the problems of a dual pair (1.17)-(1.19) and (1.20)-(1.22) will be solvable if each of these problems has at least one feasible program. This condition is necessary and sufficient.

Proof. 1. Sufficiency is established without referring to the first duality theorem.

Let $Y' = (y_1', y_2', \ldots, y_m')$ be a feasible program of problem (1.20)-(1.22). In this case we have, from Lemma 1.1 for any feasible program $X = (x_1, x_2, \ldots, x_n)$ of problem (1.17)-(1.19),

$$\sum_{j=1}^{n} c_j x_j \leqslant \sum_{i=1}^{m} b_i y_i'.$$

Thus, the set of feasible programs of problem (1.17)-(1.19) is nonempty and the linear form (1.17) is bounded from above on this set. Therefore, according to Chapter 2, Theorem 4.4, problem (1.17)-(1.19) is solvable.

The solvability of problem (1.20)-(1.22) is established analogously.

2. Necessity follows from the first duality theorem. Indeed, if one of the problems of the dual pair is solvable the other problem is solvable too. Hence, each of the two dual problems has at least one program.

Corollary 3.2. One of the problems of a dual pair has feasible programs and the set of feasible programs of the other problem is empty if and only if the linear form of the first problem is unbounded on the set of its feasible programs.

Proof. 1. Sufficiency is given explicitly by Lemma 1.3.

2. To establish necessity, we refer to the first duality theorem. If the set of feasible programs of one of the two dual problems is empty, this problem is unsolvable and, consequently, the other problem has no solutions either. By assumption, however, the latter has feasible programs. Hence, its unsolvability is due to the unboundedness of the linear form in the set of feasible programs (see Chapter 2, Theorem 4.4).

Dual pairs fall into the following three disjoint classes:

(a) the two problems have feasible programs;

(b) only one of the problems has feasible programs;

(c) for each problem of the dual pair the set of feasible programs is empty.

We give some illustrative examples.

Example 1. The primal problem I′:

$$\begin{gathered}5x_1+x_2-\max;\\ x_1+2x_2\leqslant 4,\\ 2x_1+3x_2\leqslant 3,\\ x_1\geqslant 0,\ x_2\geqslant 0.\end{gathered}$$

The dual problem II′:

$$\begin{gathered}4y_1+3y_2-\min;\\ y_1+2y_2\geqslant 5,\\ 2y_1+3y_2\geqslant 1,\\ y_1\geqslant 0,\ y_2\geqslant 0.\end{gathered}$$

Each of these problems has, obviously, feasible programs:

the vector (0, 0) is a feasible program of problem I′;

the vector (1, 2) is a feasible program of problem II′.

Example 2. The primal problem I″:

$$\begin{gathered}5x_1+x_2-\max;\\ x_1-2x_2\leqslant 2,\\ x_1-3x_2\leqslant 3,\\ x_1\geqslant 0,\ x_2\geqslant 0.\end{gathered}$$

The dual problem II″:

$$\begin{gathered}2y_1+3y_2-\min;\\ y_1+y_2\geqslant 5,\\ -2y_1-3y_2\geqslant 1,\\ y_1\geqslant 0,\ y_2\geqslant 0.\end{gathered}$$

The vector (0, 0) is a feasible program of problem I″. Problem II″ has no feasible programs. Indeed, multiplying the first restraint of problem II″ by 2 and adding it to the second restraint, we obtain

$$-y_2\geqslant 11,$$

or

$$y_2\leqslant -11,$$

which contradicts the last restraint of the problem: $y_2\geqslant 0$.

Example 3. The primal problem I‴:

$$\begin{gathered}5x_1+x_2-\max;\\ x_1-x_2\leqslant 1,\\ -x_1+x_2\leqslant -2,\\ x_1\geqslant 0,\ x_2\geqslant 0.\end{gathered}$$

The dual problem II‴:

$$\begin{gathered}y_1-2y_2-\min;\\ y_1-y_2\geqslant 5,\\ -y_1+y_2\geqslant 1,\\ y_1\geqslant 0,\ y_2\geqslant 0.\end{gathered}$$

Adding the first two restraints of problem I‴ we have

$$0\leqslant -1.$$

The ensuing contradiction shows that problem I‴ has no feasible programs. Proceeding similarly with restraints of problem II‴, we obtain $0\geqslant 6$. Hence, problem II‴ has no feasible programs either.

Thus, each of the three possibilities (a), (b), and (c) exists in practice. It follows from Corollary 3.1 that the conditions of (a) are equivalent to the assumption that the two problems of the dual pair are solvable. Referring to Corollary 3.2, we observe that (b) is tantamount to the requirement that the linear form of one of the problems be unbounded in the set of its feasible programs.

We now deal with another corollary of the first duality theorem which gives the necessary and sufficient conditions for optimality of programs of conjugate problems.

Corollary 3.3. The programs $X^*=(x_1^*, x_2^*, \ldots, x_n^*)$ *and* $Y^*=(y_1^*, y_2^*, \ldots, y_m^*)$ *of problems (1.17)–(1.19) and (1.20)–(1.22), respectively, are optimal if and only if*

$$\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} b_i y_i^*.$$

Proof. Sufficiency follows from Lemma 1.2. Necessity is given explicitly in the second part of the first duality theorem.

3-4. For the following we shall require some new definitions.

Consider the inequality restraints of problems (1.1)–(1.3) and (1.4)–(1.5). These are the inequalities (1.3) of the primal problem and the relationships (1.5) of the dual problem.

The j-th restraint in (1.5) $(\sum_{i=1}^{m} a_{ij} y_i \geqslant c_j)$ is dual with respect to the j-th restraint in (1.3) $(x_j \geqslant 0)$.

The j-th restraint in (1.3) (or (1.5)) is termed fixed if for any optimal program X^* (Y^*) of problem (1.1)–(1.3) (problem (1.4)–(1.5))

$$x_j^* = 0 \quad \left(\sum_{i=1}^{m} a_{ij} y_i^* = c_j\right),$$

i. e., if this restraint is reduced to an equality by any optimal program of the corresponding problem.

The j-th restraint in (1.3) (or (1.5)) is termed free if for at least one optimal program X^* (Y^*) of problem (1.1)–(1.3) (problem (1.4)–(1.5))

$$x_j^* > 0 \quad \left(\sum_{i=1}^{m} a_{ij} y_i^* > c_j\right),$$

i. e., if this condition is reduced to a strict inequality for at least one optimal program of the corresponding problem.

The following theorem establishes a relationship between the restraints (1.3) and the dual restraints (1.5).

Assume that problem (1.1)–(1.3) and, consequently, also the dual problem (1.4)–(1.5) are solvable.

Theorem 3.3. If some restraint in (1.3) is free (fixed), the dual restraint in (1.5) is fixed (free).

Proof. We shall use the notations introduced in the proof of Theorem 3.1.

1. We first show that if the j_0-th restraint in (1.3) is free, the j_0-th restraint in (1.5) is fixed.

We may reason geometrically thus: the optimal program of the dual problem corresponds to a hyperplane Π^* passing through the highest point of intersection of the cone K and the line Q (the point $\bar{B}$) and lying above the cone K.

Since the j_0-th restraint in (1.3) is free, the vector $\bar{B}$ may be represented as a nonnegative linear combination of the vectors $\bar{A}_j$, the vector $\bar{A}_{j_0}$ appearing in the combination with a positive coefficient. The vector A_{j_0} therefore belongs to the hyperplane Π^*, which, geometrically, indicates that the j_0-the restraint in (1.5) is fixed.

2. We now prove the first part of the theorem analytically. Consider an arbitrary solution

$$Y^* = (y_1^*, y_2^*, \ldots, y_m^*)$$

of the dual problem (1.4)–(1.5). If we take

$$\Lambda=(-y_1^*, -y_2^*, \ldots, -y_m^*, 1),$$

then

$$(\Lambda, \bar{A}_j)\leqslant 0,\ j=1,\ 2,\ \ldots,\ n, \tag{3.9}$$

since Y^* is a program of problem (1.4)–(1.5).

By assumption, there exists a solution

$$X^*=(x_1^*, x_2^*, \ldots, x_n^*)$$

of problem (1.1)–(1.3) such that $x_{j_0}^*>0$. Since Y^* and X^* are optimal programs, we have (Corollary 3.3)

$$\Delta=\sum_{j=1}^{n} c_j x_j^*=\sum_{i=1}^{m} b_i y_i^*,$$

or, equivalently,

$$(\Lambda, \bar{B})=0. \tag{3.10}$$

Substituting (3.2) into (3.10), we obtain:

$$\sum_{j=1}^{n} x_j^*(\Lambda, \bar{A}_j)=0. \tag{3.11}$$

According to (3.9) each term in the left-hand side of (3.11) is negative, thus (3.11) is satisfied only if all the terms vanish. In particular,

$$x_{j_0}^*(\Lambda, \bar{A}_{j_0})=0.$$

By assumption, however, $x_{j_0}^*>0$. Hence

$$(\Lambda, \bar{A}_{j_0})=-\sum_{i=1}^{m} y_i^* a_{ij_0}+c_{j_0}=0. \tag{3.12}$$

Since (3.12) applies for any optimal program Y^* of problem (1.4)–(1.5), the j_0-th restraint in (1.5) is obviously fixed.

3. We now prove the second part of the theorem: if the j_0-th restraint in (1.3) is fixed, the j_0-th restraint in (1.5) is free.

Again, we first give a geometrical proof. From point $\bar{B}$ — the geometrical image of the solution of problem (1.1)–(1.3) — draw two vectors: the vector e_{m+1}, directed upward along the line Q, and the vector $\bar{B}-\bar{A}_{j_0}$, drawn from $\bar{A}_{j_0}$ to $\bar{B}$. According to the conditions of the theorem, the cone spanned by and $\bar{B}-\bar{A}_{j_0}$ with apex at $\bar{B}$ satisfies all the requirements of Lemma 2.3. Therefore, through $\bar{B}$ we can draw a support hyperplane Π^* of K such that all the points of the cone, except the apex $\bar{B}$, are above Π^*. The hyperplane Π^* is the geometrical image of the optimal program of problem (1.4)–(1.5) which reduces the j_0-th restraint in (1.5) to a strict inequality.

4. We now give a rigorous proof of the second part of the theorem. Let the j_0-th restraint in (1.3) be fixed. By definition, the j_0-th component in each optimal program of problem (1.1)–(1.3) is zero. In other words, the vector $\bar{A}_{j_0}$ does not enter any of the representations (3.2) of the vector $\bar{B}$.

Consider a convex polyhedral cone $\mathrm{T}_{\bar{B}}$ with apex at $\bar{B}$ spanned by the vectors e_{m+1} and $\bar{B}-\bar{A}_{j_0}$. Let

$$R=\bar{B}+\mu_1(\bar{B}-\bar{A}_{j_0})+\mu_2 e_{m+1}, \tag{3.13}$$

$\mu_1\geqslant 0$, $\mu_2\geqslant 0$ be a point of $\mathrm{T}_{\bar{B}}$. We shall show that when $\mu_1+\mu_2>0$, $R \notin K$. For $\mu_1=0$ this was established in the proof of Theorem 3.1.

Now let $\mu_1>0$. Assuming the contrary,

$$R=\bar{B}+\mu_1(\bar{B}-\bar{A}_{j_0})+\mu_2 e_{m+1}\in K,$$

i. e.,

$$R=\sum_{j=1}^{n} x'_j\bar{A}_j, \tag{3.14}$$

where $x'_j\geqslant 0$ for $j=1, 2, \ldots, n$. Take the vector

$$R'=\frac{\mu_1}{1+\mu_1}\bar{A}_{j_0}+\frac{1}{1+\mu_1}R. \tag{3.15}$$

Since the cone K is a convex set, $R'\in K$

Substituting (3.13) into (3.15), we obtain

$$R'=\bar{B}+\frac{\mu_2}{1+\mu_1}e_{m+1}\in K.$$

If $\mu_2>0$, this relationship is contradictory to the assumption that Δ (the $(m+1)$-th component of the vector $\bar{B}$) is the maximum of the linear form (1.1) subject to the conditions (1.2) and (1.3) (see proof of Theorem 3.1). If $\mu_2=0$, however,

$$\bar{B}=R'=\frac{\mu_1}{1+\mu_1}\bar{A}_{j_0}+\frac{1}{1+\mu_1}R,$$

whence, applying (3.14), we obtain (3.2) containing A_{j_0} with a positive coefficient, which is a priori impossible.

Assumption (3.14) is thus false. Hence, $R\notin K$ if the coefficients $\mu_1\geqslant 0$ and $\mu_2\geqslant 0$ appearing in (3.13) satisfy the conditions

$$\mu_1+\mu_2>0.$$

Thus, all the points of the cone $\mathrm{T}_{\bar{B}}$ except the apex $\bar{B}$ lie outside the cone K.

Observe that $\bar{B}$ is the point of cone $\mathrm{T}_{\bar{B}}$. Indeed, for any $\mu_1\geqslant 0$, $\mu_2\geqslant 0$ it follows from the condition $\mu_1+\mu_2>0$ that

$$R=\bar{B}+\mu_1(\bar{B}-\bar{A}_{j_0})+\mu_2 e_{m+1}\neq\bar{B}\in K.$$

Therefore

$$\mu_1(\bar{B}-A_{j_0})+\mu_2 e_{m+1}\neq 0.$$

5. All the aforegoing justifies the application of Lemma 2.3 to cones K and $\mathrm{T}_{\bar{B}}$. In this case

$$P=\bar{B};\quad N_2=2;\quad \mathrm{T}_1=\bar{B}-\bar{A}_{j_0},\quad \mathrm{T}_2=e_{m+1}.$$

According to Lemma 2.3, there exists a hyperplane Π^* with the equation

$$(\Lambda, U)=0,$$

satisfying the conditions

$$(\Lambda, \bar{A}_j)\leqslant 0,\qquad j=1, 2, \ldots, n; \tag{3.16}$$

$$(\Lambda, \bar{B})=0; \tag{3.17}$$

$$(\Lambda, \bar{B}+e_{m+1})=(\Lambda, e_{m+1})>0; \tag{3.18}$$

$$(\Lambda, \bar{B}+(\bar{B}-\bar{A}_{j_0}))=-(\Lambda, \bar{A}_{j_0})>0. \tag{3.19}$$

Inequalities (3.16) and equality (3.17) indicate that Π^* is the support hyperplane of the cone K at the point $\bar{B}$. Relationships (3.18) and (3.19) follow from that property of the hyperplane Π^* according to which all the points of the cone $\mathrm{T}_{\bar{B}}$, except the apex, are above Π^*.

According to (3.18) the $(m+1)$-th component of the vector $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m, \lambda_{m+1})$ is positive. We may thus introduce a vector $Y^*=(y_1^*, y_2^*, \ldots, y_m^*)$ taking

$$y_i^*=-\frac{\lambda_i}{\lambda_{m+1}}, \qquad i=1, 2, \ldots, m.$$

Applying conditions (3.16) and (3.17) we conclude, as in the proof of Theorem 3.1, that the vector Y^* is a solution of the dual problem (1.4)–(1.5). According to inequality (3.19)

$$\sum_{i=1}^{m}\lambda_i a_{ij_0}+\lambda_{m+1}c_{j_0}<0,$$

or, equivalently,

$$\sum_{i=1}^{m}\left(-\frac{\lambda_i}{\lambda_{m+1}}\right)a_{ij_0}=\sum_{i=1}^{m}y_i^*a_{ij_0}>c_{j_0}. \tag{3.20}$$

We have thus derived an optimal program Y^* of problem (1.4)–(1.5) satisfying inequality (3.20). By definition, this indicates that the j_0-th restraint in (1.5) is free. This completes the proof.

Consider a linear-programming problem with homogeneous restraints (1.17)–(1.19) and the dual problem (1.20)–(1.22).

We reduce problem (1.17)–(1.19) to canonical form introducing additional variables

$$x_{n+i}=b_i-\sum_{j=1}^{n}a_{ij}x_j\geqslant 0, \qquad i=1, 2, \ldots, m.$$

We may now extend the definitions introduced at the beginning of this section to problems with homogeneous restraints. In this case the restraint vectors A_j have the form;

$$A_j=\begin{cases}(a_{1j}, a_{2j}, \ldots, a_{mj})^T, & \text{if } 1\leqslant j\leqslant n;\\ (\underbrace{0, \ldots, 0}_{n-j}, 1, 0, \ldots, 0)^T, & \text{if } n+1\leqslant j\leqslant n+m.\end{cases}$$

Therefore the restraint

$$(A_j, Y)=\sum_{i=1}^{m}a_{ij}y_i\geqslant c_j \qquad (j=1, 2, \ldots, n)$$

of problem (1.20)–(1.22) is dual with respect to the restraint $x_j\geqslant 0$ of problem (1.17)–(1.19). Analogously, the restraint

$$(A_{n+i}, Y)=y_i\geqslant 0 \qquad (i=1, 2, \ldots, m)$$

of problem (1.20)–(1.22) is dual with respect to the restraint

$$x_{n+i}=b_i-\sum_{j=1}^{n}a_{ij}x_j\geqslant 0$$

of problem (1.17)–(1.19).

The j-th restraint in (1.21) is thus dual with respect to the j-th restraint in (1.19); the i-th restraint in (1.22) is dual with respect to the i-th restraint in (1.18).

If problem (1.20)–(1.22) is taken as the primal problem then, as shown in § 1, problem (1.17)–(1.19) is the dual problem. Hence, the j-th restraint in (1.19) (the i-th restraint in (1.18)) is dual with respect to the j-th restraint in (1.21) (i-th restraint in (1.22)). The j-th restraints in (1.19), (1.21)

$$x_j\geqslant 0, \quad \sum_{i=1}^{m}a_{ij}y_i\geqslant c_j \tag{3.21}$$

and the i-th restraints in (1.18), (1.22)

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i, \quad y_i \geqslant 0 \tag{3.22}$$

are thus pairs of dual restraints of the conjugate problems (1.17)-(1.19) and (1.20)-(1.22). Observe that the dual pair (3.21) corresponds to the j-th column in Table 3.1, and its component restraints are therefore termed column restraints.

Analogously, the components of the dual pair (3.22) are termed row restraints (they correspond to the i-th row of Table 3.1).

We now extend the definition of the fixed and free restraints given at the beginning of this section to problems with homogeneous restraints. A restraint is said to be free if there exists a solution reducing the restraint to a strict inequality; a restraint is said to be fixed if it is reduced to an equality for all the solutions of the given problem.

After these preliminary remarks we can formulate the second theorem for problems with homogeneous restraints.

Theorem 3.4. (Second duality theorem.) If the conjugate problems (1.17)-(1.19) and (1.20)-(1.22) are solvable, then in each pair of their dual restraints (either row or column) one restraint is free and the other is fixed.

The second duality theorem is an obvious consequence of Theorem 3.3 and of the definitions for problems with homogeneous restraints.

3-5. The first and the second duality theorems will be used repeatedly in the following. It is therefore important that their actual content be completely understood. The propositions of the duality theorems become clear if they are restated in terms of production of a homogeneous commodity. These terms have already been used in the economic interpretation of the pair of dual problems with identical restraints.

According to the first duality theorem, an optimal production program exists only if all the production factors have evaluations. Suppose that the production has an optimal program and, consequently, all the factors can be evaluated. In reality this is not always so. Both the optimal production program and the system of costing of the production factors are not usually uniquely determinable.

According to the second part of the first duality theorem, the cost of the commodity produced under some optimal production program is equal to the total cost of the resources available, regardless of the actual evaluation of the production factors (the components of the solution of the dual problem). Thus, a characteristic property of the optimal program is that it ensures that the cost evaluation of the commodity produced and that of the resources consumed in production are equal. With any other program of utilization of the technological means (that is, other than optimal) the production process will be deficient: the cost of the commodity produced will be less than the total cost of the resources available. This is due to the fact that under a nonoptimal program the possibilities of production are not fully exploited.

We recall that a cost vector of production factors is taken as the set of preliminary evaluations of production factors (a program of the dual problem) which minimizes the total evaluation of the resources available (the linear form of the dual problem). The meaning of this now becomes clear. When this condition is observed, the characteristic property of the

optimal program is fulfilled. If, however, the cost vector is taken as some set of preliminary evaluations not minimizing the total cost of the resources available, the characteristic property of the optimal program does not apply.

In this case the cost of the commodity produced under any program, including an optimal program, is less than the total cost of the resources available. The costs culminating in this situation are obviously undesirable.

We shall now proceed with the economic interpretation of the second duality theorem.

We first consider the row restraints of the dual pair of problems defining a given production process.

Suppose that under one of the optimal programs of utilization of technological facilities the i-th production factor is not fully exploited. Then, according to the second duality theorem, its cost vanishes. Conversely, if the cost of an i-th production factor (with any cost vector) is zero, there exists an optimal production program under which the resources of the i-th factor are not fully exploited.

This situation can be regarded as quite natural.

Indeed, a factor whose resources exceed the requirements (from the viewpoint of some optimal production program) is not critical in the production process: a certain reduction of the resources of this factor will not limit the scope of production. The cost of this factor from the viewpoint of the given production process can naturally be taken, therefore, as zero. Now suppose that the excess of the particular factor has been eliminated. In this case the factor acquires a certain value: further reduction of the resources will affect the volume of the output commodity. The cost of the factor thus becomes positive.

We now give an economic interpretation of the column restraints of the dual pair of problems.

According to the second duality theorem, a given production process is foreseen by some optimal program only if the resulting cost of the commodity produced is equal to the cost of the resources consumed (from the viewpoint of any cost vector). The intuitive content of this proposition is quite obvious: there is no point in following a production process in which the costs incurred will exceed the potential income.

§ 4. Linear-programming problems in arbitrary form

4-1. In § 3 we established duality relationships for linear-programming problems with homogeneous restraints. In this section we propose to extend these results to linear-programming problems given in arbitrary form.

We formulate a general problem of this type.

Maximize the linear form

$$L(X) = \sum_{j=1}^{n} c_j x_j \tag{4.1}$$

subject to the conditions

$$\sum_{j=1}^{n} a_{ij} x_j \begin{cases} \leqslant b_i, & i = 1, 2, \ldots, m_1 \leqslant m, \\ = b_i, & i = m_1 + 1, m_1 + 2, \ldots, m; \end{cases} \tag{4.2}$$

$$x_j \geqslant 0, \quad j=1, 2, \ldots, n_1 \leqslant n. \tag{4.3}$$

One group of restraints (4.2) imposed on all the variables of the problem are inequalities (numbering m_1), whereas the other group consists of equalities (the number of equalities being $m-m_1$). Moreover, some of the variables x_j are assumed to be nonnegative (n_1 variables). The fact that it is the first m_1 restraints (4.2) that are inequalities and that constraints (4.3) are imposed on the first n_1 variables does not detract from the generality of the statement of the problem. This arrangement can always be changed by suitably renumbering the variables and the restraints of the problem. The restraints of problem (4.1)–(4.3) are obviously mixed (as opposed to the homogeneous restraints of problem (1.17)–(1.19)). Problems of type (4.1)–(4.3) are therefore called problems with mixed restraints

By definition, the dual (or the conjugate) problem with respect to problem (4.1)–(4.3) consists in minimizing the linear form

$$\sum_{i=1}^{m} b_i y_i \tag{4.4}$$

subject to the conditions

$$\sum_{i=1}^{m} a_{ij} y_i \begin{cases} \geqslant c_j, & j=1, 2, \ldots, n_1 \leqslant n, \\ = c_j, & j=n_1+1, n_1+2, \ldots, n; \end{cases} \tag{4.5}$$

$$y_i \geqslant 0, \quad i=1, 2, \ldots, m_1 \leqslant m. \tag{4.6}$$

The dual problem of a problem with mixed restraints (4.1)–(4.3) is thus formed according to the following rules.

If the variable x_j of problem (4.1)–(4.3) is assumed nonnegative, the j-th restraint in (4.5) is an inequality. If, however, no such restraint is imposed on x_j, the j-th relationship in (4.5) is an equality. Analogous correspondence is established between restraints (4.2) of problem (4.1)–(4.3) and constraints (4.6) of problem (4.4)–(4.6). If the i-th restraint in (4.2) is an inequality, $y_i \geqslant 0$. Otherwise, the variable y_i may take on either positive or negative values.

It can easily be verified (see Exercise 10) that problem (4.1)–(4.3) is dual with respect to problem (4.4)–(4.5).

These two problems may therefore be designated as a pair of dual (or conjugate) problems.

Observe that the preceding definition of a dual problem does not contradict the concept of duality introduced elsewhere for problems with homogeneous restraints and for canonical problems.

If we take $m_1=m$, $n_1=n$ problem (4.1)–(4.3) reduces to a problem with homogeneous restraints (1.17)–(1.19), and the dual problem (4.4)–(4.6) transforms to problem (1.20)–(1.22). For $m_1=0$, $n_1=n$, problems (4.1)–(4.3) and (4.4)–(4.6) reduce to problems (1.1)–(1.3) and (1.4)–(1.5), respectively.

4-2. Each linear-programming problem of type (4.1)–(4.3) will be associated with the following problem with homogeneous restraints.

Maximize the linear form

$$\sum_{j=1}^{n_1} c_j x_j' + \sum_{j=n_1+1}^{n} c_j (x_j' - x_{j+n_1}') \tag{4.7}$$

subject to the conditions

$$\sum_{j=1}^{n_1} a_{ij} x_j' + \sum_{j=n+1}^{n} a_{ij} (x_j' - x_{j+n_1}') \leqslant b_i, \tag{4.8}$$
$$i=1, 2, \ldots, m,$$

$$-\sum_{j=1}^{n_1} a_{ij}x'_j - \sum_{j=n_1+1}^{n} a_{ij}(x'_j - x'_{j+n_2}) \leq -b_i,$$
$$i = m_1+1,\ m_1+2,\ \dots,\ m; \tag{4.8}$$

$$x'_j \geq 0,\quad j = 1,\ 2,\ \dots,\ n,\ n+1,\ \dots,\ n+n_2, \tag{4.9}$$

where $n_2 = n - n_1$ is the number of variables in problem (4.1)–(4.3) which are not assumed nonnegative.

We establish a correspondence between problems (4.1)–(4.3) and (4.7)–(4.9).

An n-dimensional vector $X = (x_1, x_2, \dots, x_n)$ and an $(n+n_2)$-dimensional vector $X' = (x'_1, x'_2, \dots, x'_{n+n_2})$ will be said to be c o r r e s p o n d i n g if their components are related by the expressions

$$x_j = \begin{cases} x'_j, & j = 1,\ 2,\ \dots,\ n_1, \\ x'_j - x'_{j+n_2}, & j = n_1+1,\ n_1+2,\ \dots,\ n. \end{cases} \tag{4.10}$$

To any $(n+n_2)$-dimensional vector X' there obviously corresponds a unique n-dimensional vector X. Conversely, to any n-dimensional vector X there corresponds an entire family of $(n+n_2)$-dimensional vectors X'.

The correspondence established by (4.10) is thus single-valued in one direction only.

Let $X' = (x'_1, x'_2, \dots, x'_{n+n_2})$ be a feasible program of problem (4.7)–(4.9). Applying (4.10), we observe that the corresponding vector $X = (x_1, x_2, \dots, x_n)$ is a feasible program of problem (4.1)–(4.3).

Indeed,

$$\sum_{j=1}^{n_1} a_{ij}x'_j + \sum_{j=n_1+1}^{n} a_{ij}(x'_j - x'_{j+n_2}) = \sum_{j=1}^{n} a_{ij}x_j. \tag{4.11}$$

Applying, further, conditions (4.8), we obtain

$$\sum_{j=1}^{n} a_{ij}x_j \begin{cases} \leq b_i, & i = 1,\ 2,\ \dots,\ m, \\ \geq b_i, & i = m_1+1,\ \dots,\ m, \end{cases}$$

or

$$\sum_{j=1}^{n} a_{ij}x_j \begin{cases} \leq b_i, & i = 1,\ 2,\ \dots,\ m_1, \\ = b_i, & i = m_1+1,\ m_1+2,\ \dots.\ m. \end{cases}$$

Moreover, with $j = 1, 2, \dots, n_1$, $x_j = x'_j \geq 0$.

The vector $X = (x_1, x_2, \dots, x_n)$ thus satisfies restraints (4.2), (4.3) of problem (4.1)–(4.3) and as such is a feasible program.

Let now $X = (x_1, x_2, \dots, x_n)$ be a feasible program of problem (4.1)–(4.3).

Among the $(n+n_2)$-dimensional vectors X' corresponding to X a priori there exist vectors with nonnegative components. One of these is the vector $\overline{X}' = (\overline{x}'_1, \overline{x}'_2, \dots, \overline{x}'_{n+n_2})$ where

$$\overline{x}'_j = \begin{cases} x_j, & j = 1,\ 2,\ \dots,\ n_1, \\ \max(0,\ x_j), & j = n_1+1,\ n_2+2,\ \dots,\ n, \\ \max(0,\ -x_j), & j = n+1,\ n+2,\ \dots,\ n+n_2. \end{cases} \tag{4.12}$$

The nonnegativity of all the components of $\overline{X}'$ is obvious. The correspondence of the vectors X and $\overline{X'}$ follows from the equalities

$$\overline{x}'_j - \overline{x}'_{j+n_2} = \max(0,\ x_j) - \max(0,\ -x_j) = \begin{cases} x_j - 0 \quad = x_j, & \text{if } x_j \geq 0, \\ 0 - (-x_j) = x_j, & \text{if } x_j < 0, \end{cases}$$

which apply for $j = n_1+1,\ n_1+2,\ \dots,\ n$.

It follows from (4.11) and from (4.2), (4.3) that any vector X' with

nonnegative components corresponding to a feasible program X of problem (4.1)–(4.3) is a feasible program of problem (4.7)–(4.9).

To sum up, to any feasible program of problem (4.7)–(4.9) corresponds a feasible program of problem (4.1)–(4.3). Conversely, to any feasible program of problem (4.1)–(4.3) corresponds some set of feasible programs of problem (4.7)–(4.9). This correspondence is established by (4.10).

We shall show that the linear forms of problems (4.1)–(4.3) and (4.7)–(4.9) take on the same values on the corresponding programs X and X' of these problems.

Indeed, applying (4.10), we obtain

$$\sum_{j=1}^{n} c_j x_j = \sum_{j=1}^{n_1} c_j x_j' + \sum_{j=n_1+1}^{n} c_j (x_j' - x_{j+n_2}').$$

Hence, it follows that (4.10) also establishes a correspondence between the solutions of problems (4.1)–(4.3) and (4.7)–(4.9).

Problem (4.4)–(4.6), the dual problem of (4.1)–(4.3), can easily be reduced to the form (4.1)–(4.3). It suffices to take $\bar{c}_j = -c_j$, $\bar{a}_{ij} = -a_{ij}$, $\bar{b}_i = -b_i$. With this substitution, problem (4.4)–(4.6) reduces to maximization of the linear form

$$\sum_{i=1}^{m} \bar{b}_i y_i$$

subject to the conditions

$$\sum \bar{a}_{ij} y_i \begin{cases} \leqslant \bar{c}_j, & j = 1, 2, \ldots, n_1, \\ = \bar{c}_j, & j = n_1 + 1, \ldots, n, \end{cases}$$
$$y_i \geqslant 0, \quad i = 1, 2, \ldots, m_1.$$

To problem (4.4)–(4.6), therefore, corresponds the following problem with homogeneous restraints.

Minimize the linear form

$$\sum_{i=1}^{m_1} b_i y_i' + \sum_{i=m_1+1}^{m} b_i (y_i' - y_{i+m_2}') \tag{4.13}$$

subject to the conditions

$$\left.\begin{aligned} &\sum_{i=1}^{m_1} a_{ij} y_i' + \sum_{i=m_1+1}^{m} a_{ij} (y_i' - y_{i+m_2}') \geqslant c_j, \\ &\qquad j = 1, 2, \ldots, n, \\ &-\sum_{i=1}^{m_1} a_{ij} y_i' - \sum_{i=m_1+1}^{m} a_{ij} (y_i' - y_{i+m_2}') \geqslant -c_j, \\ &\qquad j = n_1 + 1, \ldots, n, \end{aligned}\right\} \tag{4.14}$$

$$y_i' \geqslant 0, \quad i = 1, 2, \ldots, m, m+1, \ldots, m+m_2, \tag{4.15}$$

where $m_2 = m - m_1$ is the number of variables in problem (4.4)–(4.6) which are not constrained by the assumption of nonnegativity.

According to (4.10), an m-dimensional vector $Y = (y_1, y_2, \ldots, y_m)$ and an $(m+m_2)$-dimensional vector $Y' = (y_1', y_2', \ldots, y_{m+m_2}')$ are said to correspond if

$$y_i = \begin{cases} y_i', & i = 1, 2, \ldots, m_1, \\ y_i' - y_{i+m_2}', & i = m_1 + 1, \ldots, m. \end{cases} \tag{4.16}$$

We have proved that to any program Y' of problem (4.13)–(4.15) corresponds a feasible program Y of problem (4.4)–(4.6). Conversely, any vector Y' with nonnegative components corresponding to a feasible program Y of problem (4.4)–(4.6) is a feasible program of problem (4.13)–(4.15). Moreover, if Y and Y' are two corresponding programs of problems (4.4)–(4.6) and (4.13)–(4.15), the optimality of either program yields the optimality of the other.

4-3. We are now ready to extend duality theorems to problems with mixed restraints.

Theorem 4.1. (First duality theorem: general case.) If one of the problems of a dual pair (4.1)-(4.3) and (4.4)-(4.6) is solvable, the other problem is also solvable. Any optimal programs $X=(x_1, x_2, \ldots, x_n)$ *and* $Y=(y_1, y_2, \ldots, y_m)$ *of these problems satisfy the equality*

$$\sum_{j=1}^{n} c_j x_j = \sum_{i=1}^{m} b_i y_i. \tag{4.17}$$

Proof. Assume that problem (4.1)-(4.3) is solvable and let X be its optimal program. Consider the vector $X'=(x_1', x_2', \ldots, x_{n+n_2}')$ associated with X by (4.10) and having nonnegative components. According to the above, X' is a solution of (4.7)-(4.9).

We now apply the first duality theorem for problems with homogeneous restraints (Theorem 3.2) according to which problem (4.13)-(4.15), the dual of problem (4.7)-(4.9) is solvable and its optimal program $Y'=(y_1', y_2', \ldots, y_{m+m_2}')$ satisfies the equality

$$\sum_{j=1}^{n_1} c_j x_j' + \sum_{j=n_1+1}^{n} c_j (x_j' - x_{j+n_2}') = \sum_{i=1}^{m_1} b_i y_i' + \sum_{i=m_1+1}^{m} b_i (y_i' - y_{i+m_2}'). \tag{4.18}$$

As stated before, the vector Y corresponding by (4.16) to the optimal program Y' of problem (4.13)-(4.15) is a solution of problem (4.4)-(4.6).

Further, applying the equalities

$$\sum_{j=1}^{n_1} c_j x_j' + \sum_{j=n_1+1}^{n} c_j (x_j' - x_{j+n_2}') = \sum_{j=1}^{n} c_j x_j,$$

$$\sum_{i=1}^{m_1} b_i y_i' + \sum_{i=m_1+1}^{m} b_i (y_i' - y_{i+m_2}') = \sum_{i=1}^{m} b_i y_i$$

and (4.18) we obtain (4.17) which holds for any solutions X and Y of the dual pair.

In our proof we set out from problem (4.1)-(4.3), If now we assume solvability of problem (4.4)-(4.6), the proof of the theorem is analogous. However we do not give the proof again since problem (4.1)-(4.3) is dual with respect to problem (4.4)-(4.6). This completes the proof.

The second duality theorem in the general case applies only to inequality restraints. A pair of restraints of the dual problems (4.1)-(4.3) and (4.4)-(4.6) with the same subscript $j (1 \leqslant j \leqslant n_1)$ or the same subscript $i (1 \leqslant i \leqslant m_1)$ will be called d u a l. In the first case (given j) the dual conditions are called c o l u m n r e s t r a i n t s, and in the second case (given i) they are said to be r o w r e s t r a i n t s.

The definition of free and fixed restraints for these problems is not different from the corresponding definitions for problems with homogeneous restraints. We shall therefore not repeat these definitions.

Theorem 4.2. (Second duality theorem: general case.) If the dual pair (4.1)-(4.3) and (4.4)-(4.6) is solvable, then in each pair of dual restraints (column restraints for $j=1, 2, \ldots, n_1$ *and row restraints for* $i=1, 2, \ldots, m_1$*) one restraint is free, and the other is fixed.*

Proof. Consider problems (4.7)-(4.9) and (4.13)-(4.15) related to problems (4.1)-(4.3) and (4.4)-(4.6). Assume that the i-th restraint in (4.2) of problem (4.1)-(4.3) is free ($i=1, 2, \ldots, m_1$). In this case the i-th restraint in (4.8) of problem (4.7)-(4.9) is also free.

Indeed, let X be an optimal program of problem (4.1)-(4.3) reducing the

l-th restraint in (4.2) to an inequality. If X' is a vector with nonnegative components corresponding to program X according to (4.10), then, from the above, it is a solution of problem (4.7)–(4.9). Now, applying (4.11), we may assert that X' reduces the l-th restraint in (4.8) to an inequality. Hence this is a free restraint.

Now, let the l-th restraint in (4.2) be fixed. Consider a solution X' of problem (4.7)–(4.9).

According to the above, the vector X corresponding to X' by (4.10) is an optimal program of problem (4.1)–(4.3). Hence, the l-th restraint in (4.2) is reduced to an equality by vector X. Applying (4.11), we observe that the vector X' reduces the l-th restraint in (4.8) to an equality. Our argument remains the same if row restraints are replaced by column restraints.

To sum up, to a free (fixed) restraint of problem (4.1)–(4.3) corresponds a free (fixed) restraint of problem (4.7)–(4.9).

The same correspondence between restraints obviously applies to problems (4.4)–(4.6) and (4.13)–(4.15) also.

After these preliminaries, the theorem is proved as follows.

Let some inequality restraint of problem (4.1)–(4.3) be free (fixed). The corresponding restraint of problem (4.7)–(4.9) will have the same property. Applying the second duality theorem for problems with homogeneous restraints (Theorem 3.4), we observe that the restraint of problem (4.13)–(4.15), which is dual with respect to the restraint of problem (4.7)–(4.9) under consideration, is fixed (free).

Further, applying the preceding correspondence between the restraints of problem (4.4)–(4.6) and (4.13)–(4.15), we conclude that the restraint of problem (4.4)–(4.6) dual with respect to any free (fixed) restraint of problem (4.1)–(4.3) is fixed (free). This completes the proof.

Thus, both duality theorems hold for linear-programming problems given in arbitrary form.

All the corollaries of the duality theorems which were derived in the preceding section for problems with homogeneous restraints and for canonical problems can, obviously, be extended to the general case, also. Furthermore, all three lemmas of § 1 can be extended without any modification to linear-programming problems with mixed restraints. The proofs in each case follow line by line the arguments given of § 1. In the following we shall apply these lemmas to linear-programming problems in arbitrary form.

§ 5. Optimality criteria and decision multipliers

5-1. When solving linear-programming problems, it is most important that there be some means of testing the feasible programs for optimality. In other words, we must be able to answer whether a given program is optimal.

The necessary and sufficient conditions for the optimality of a feasible program were first established by L. V. Kantorovich /61/. Later it was found that these conditions (generally called the optimality criterion of programs of a given problem) are closely connected with duality and are, essentially, consequences of the duality theorem. This section deals mainly with this connection. Such an approach enables us to apply the

results of preceding sections of this chapter and, in our opinion, makes the discussion more natural.

Consider an arbitrary problem with mixed restraints (problem (4.1)-(4.3)). Following Kantorovich /61/, we shall refer to the parameters λ_1, $\lambda_2, \ldots, \lambda_m$ as decision multipliers* if

(a) $$\sum_{i=1}^{m} a_{ij}\lambda_i \geqslant c_j, \qquad j=1, 2, \ldots, n_1; \tag{5.1}$$

(b) $$\sum_{i=1}^{m} a_{ij}\lambda_i = c_j, \qquad j=n_1+1, n_1+2, \ldots, n; \tag{5.2}$$

(c) $$\lambda_i \geqslant 0, \qquad i=1, 2, \ldots, m_1; \tag{5.3}$$

(d) for some feasible program $X=(x_1, x_2, \ldots, x_n)$ of problem (4.1)-(4.3),

$$\sum_{i=1}^{m} a_{ij}\lambda_i = c_j \text{ for } x_j>0 \ (1 \leqslant j \leqslant n_1); \tag{5.4}$$

$$\lambda_i = 0 \text{ for } \sum_{j=1}^{n} a_{ij}x_j < b_i \ (1 \leqslant i \leqslant m_1). \tag{5.5}$$

The vector $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$ whose components are the decision multipliers λ_i will be called the decision vector of problem (4.1)-(4.3) (the decision vector of program X). According to the following theorem, finding the decision vector is equivalent to solving problem (4.4)-(4.6), the dual of problem (4.1)-(4.3).

Theorem 5.1. The set of all decision vectors of problem (4.1)-(4.3) coincides with the set of optimal programs of problem (4.4)-(4.6).

Proof. 1. Let $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$ be a decision vector of problem (4.1)-(4.3), related by (5.4) and (5.5) to program $X=(x_1, x_2, \ldots, x_n)$ of this problem.

Conditions (5.1)-(5.3) satisfied by the vector Λ indicate that Λ is a feasible program of problem (4.4)-(4.6). The optimality of program Λ is thus obvious.

Let E denote the set of all subscripts j ($j=1, 2, \ldots, n_1$) for which $x_j>0$. Then

$$\sum_{j=1}^{n} c_jx_j = \sum_{j\in E} c_jx_j + \sum_{j=n_1+1}^{n} c_jx_j. \tag{5.6}$$

Applying equality (5.6), conditions (5.2) for $j=n_1+1, \ldots, n$, and (5.4) for $j\in E$, we obtain

$$\sum_{j=1}^{n} c_jx_j = \sum_{j\in E}\Big(\sum_{i=1}^{m} a_{ij}\lambda_i\Big)x_j + \sum_{j=n_1+1}^{n}\Big(\sum_{i=1}^{m} a_{ij}\lambda_i\Big)x_j.$$

By assumption, however, $x_j=0$ if $j\notin E$. Hence,

$$\sum_{j=1}^{n} c_jx_j = \sum_{j=1}^{n}\Big(\sum_{i=1}^{m} a_{ij}\lambda_i\Big)x_j = \sum_{i=1}^{m}\lambda_i\sum_{j=1}^{n} a_{ij}x_j. \tag{5.7}$$

Further, applying the equality

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \quad m_1+1 \leqslant i \leqslant m$$

and condition (5.5), we have

$$\sum_{i=1}^{m}\lambda_i\sum_{j=1}^{n} a_{ij}x_j = \sum_{i=1}^{m}\lambda_ib_i.$$

Comparing the last equality with (5.7) we obtain

$$\sum_{j=1}^{n} c_jx_j = \sum_{i=1}^{m} b_i\lambda_i. \tag{5.8}$$

* In Kantorovich's last book /65/ decision multipliers are called objectively-conditioned costs. This term follows from the economic interpretation of the solution of the dual problem.

According to Lemma 1.2, equality (5.8) proves the optimality of programs X and Λ.

Thus $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$ is a solution of problem (4.4)–(4.6).

2. Now let $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$ be a solution of problem (4.4)–(4.6).

Since the vector Λ is a feasible program of problem (4.4)–(4.6), it satisfies conditions (5.1)–(5.3).

Let E_1 denote the set of subscripts of the free restraints in (4.2) and E_2 the analogous set of subscripts in (4.3).

According to the second duality theorem, the i-th restraint in (4.6) (the j-th restraint in (4.5)) is fixed if $i \in E_1$ ($j \in E_2$). Hence

$$\lambda_i=0 \quad \text{for} \quad i \in E_1; \tag{5.9}$$

$$\sum_{i=1}^{m} a_{ij}\lambda_i=c_j \quad \text{for} \quad j \in E_2. \tag{5.10}$$

Consider a solution $X=(x_1, x_2, \ldots, x_n)$ of problem (4.1)–(4.3). If $x_j>0$, then, by definition, $j \in E_2$ and (5.10) applies. If $\sum_{j=1}^{n} a_{ij}x_j<b_i$, then $i \in E_1$ and (5.9) holds. The vector Λ thus satisfies conditions (5.1)–(5.3) and is associated by (5.4), (5.5) with some program of problem (4.1)–(4.3) (any solution of the problem may be taken as a suitable program). Hence $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$ is a decision vector of problem (4.1)–(4.3). This completes the proof.

5-2. The optimality criteria of programs of linear-programming problems are more easily stated in terms of decision multipliers.

Theorem 5.2. (Optimality criterion of programs of problem (4.1)–(4.3).) A feasible program $X=(x_1, x_2, \ldots, x_n)$ of problem (4.1)–(4-3) is optimal if and only if a decision vector $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$ exists, associated with this program by (5.4), (5.5).

Proof. Necessity. Let $X=(x_1, x_2, \ldots, x_n)$ be a solution of problem (4.1)–(4.3). According to the first duality theorem, the dual problem (4.4)–(4.6) is solvable.

Let $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$ be an optimal program of this problem. Then, according to Theorem 5.1, the vector Λ is a decision vector of problem (4.1)–(4.3), and, moreover, as has been shown in the proof of the second part of Theorem 5.1, the vector Λ is associated by (5.4), (5.5) with any solution of problem (4.1)–(4.3) and, consequently, with the solution X under consideration.

Sufficiency. Assume that a decision vector $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$ exists, associated with the given program X of problem (4.1)–(4.3) by (5.4), (5.5).

In our proof of the first part of Theorem 5.1 we established (5.8). This equality (remembering that X and Λ are feasible programs of problems (4.1)–(4.3) and (4.4)–(4.6), respectively) indicates optimality of program X (see Lemma 1.2). This completes the proof.

This criterion gives us a fairly convenient tool for deciding whether a given feasible program is a solution of the problem. The general scheme of testing programs for optimality is as follows.

Using the system of equations (5.2), (5.4), and (5.5) determine the vector $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$. Then, by substitution check whether this vector satisfies conditions (5.1), (5.3). If it does, the vector X is a solution of the problem; otherwise it is not an optimal program.

The optimality criterion is generally employed when analyzing support programs of a problem. We will deal with this aspect in more detail.

Consider a support program X of problem (4.1)-(4.3). Let

$$x_j \begin{cases} =0 & \text{for } j=1,\ 2,\ \ldots,\ n_2 \leqslant n_1, \\ >0 & \text{for } j=n_2+1,\ \ldots,\ n_1, \end{cases} \tag{5.11}$$

$$\sum_{j=1}^{n} a_{ij}x_j \begin{cases} <b_i & \text{for } i=1,\ 2,\ \ldots,\ m_2 \leqslant m_1, \\ =b_i & \text{for } i=m_2+1,\ \ldots,\ m_1, \end{cases} \tag{5.12}$$

Write out the matrix whose elements are coefficients of the restraints reduced to equalities by program X:

$$A_X = \begin{array}{l} \left(\begin{matrix} 1 & 0 & \ldots 0 & \ldots 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \ldots 1 & \ldots 0 \\ a_{m_2+1,1} & a_{m_2+1,2} & \ldots a_{m_2+1,n_2} & \ldots a_{m_2+1,n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \ldots a_{mn_2} & \ldots a_{mn} \end{matrix}\right) \begin{matrix} \left.\vphantom{\begin{matrix}1\\ \cdot \\ 0\end{matrix}}\right\} n_2 \\ \left.\vphantom{\begin{matrix}a\\ \cdot \\ a\end{matrix}}\right\} m-m_2 \end{matrix} \end{array}.$$

Since X is assumed to be a support program, there are A_X linearly independent rows among the $m+n_2-m_2$ rows of matrix n. The unit vectors constituting the first A_X rows of matrix n_2 are linearly independent. Of the remaining $m-m_2$ rows we may, therefore, choose $n-n_2$ rows which, together with the first n_2 rows of matrix A_X, constitute a linearly independent system.

Write this linearly independent system as a determinant of order n and expand it in the first n_2 rows. A nonzero determinant of order $n-n_2$ whose rows are some of the vectors of the system

$$(a_{i,n_2+1},\quad a_{i,n_2+2},\ \ldots,\ a_{i,n}),\quad i=m_2+1,\ \ldots,\ m, \tag{5.13}$$

is obtained. Hence the system (5.13) of $(n-n_2)$-dimensional vectors is of rank $n-n_2$.

If we assume that the program X is nondegenerate, the number of rows in A_X is equal to the number of columns, i. e.,

$$m-m_2+n_2=n.$$

In this case the number of vectors, $m-m_2$, in system (5.13) coincides with the rank of the system, $n-n_2$.

Thus, if X is a nondegenerate program, (5.13) comprises $n-n_2$ linearly independent $(n-n_2)$-dimensional vectors. If, however, X is a degenerate program, the number of vectors in system (5.13) is greater than its rank.

Following these preliminary remarks we now test program X for optimality. We shall try to find a decision vector $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$ associated with this program by (5.4), (5.5). Applying (5.4) and (5.11), we obtain

$$\sum_{i=1}^{m} \lambda_i a_{ij}=c_j \quad \text{for} \quad j=n_2+1,\ n_2+2,\ \ldots,\ n.$$

On the other hand, according to (5.5) and (5.12),

$$\lambda_i=0 \quad \text{for} \quad i=1,2,\ \ldots,\ m_2.$$

Hence, the remaining $m-m_2$ components of the decision vector Λ (if it exists) must satisfy the system of equations

$$\sum_{i=m_2+1}^{m} \lambda_i a_{ij}=c_j,\quad j=n_2+1,\ \ldots,\ n. \tag{5.14}$$

The columns of the matrix of coefficients of (5.14) are the vectors (5.13).

The rank of system (5.14) is, therefore, $n-n_2$, i.e., equal to the number of equations in this system.

The analysis of program X proves to be very simple especially in dealing with a nondegenerate program. In this case the number of unknowns in (5.14) is equal to the number of equations in the system and there exists a unique solution. Solving the system, we obtain a unique vector Λ which satisfies conditions (5.2), (5.4), and (5.5). We now substitute the vector Λ into the left-hand sides of (5.1), (5.3), which in this case have the form

$$\left.\begin{aligned} \sum_{i=m_2+1}^{m} a_{ij}\lambda_i \geqslant c_j, \quad & j=1, 2, \ldots, n_2, \\ \lambda_i \geqslant 0, \quad & i=m_2+1, \ldots, m_1. \end{aligned}\right\} \tag{5.15}$$

If all the conditions (5.15) are satisfied, Λ is a decision vector. According to the optimality criterion, this proves that the program X associated with Λ by (5.4), (5.5) solves the problem in question.

If, however, at least one of relationships (5.15) does not hold, no decision vector for program X exists and, consequently, this is not an optimal program. Observe that the last proposition is a consequence of the existence of a unique solution of system (5.14).

The situation is more complicated if X is a degenerate program. In this case the number of unknowns in (5.14) exceeds the number of equations and the system has an infinite number of solutions. According to the optimality criterion, X is an optimal program if and only if at least one solution of system (5.14) satisfies conditions (5.15).

Let us solve (5.14) for some $n-n_2$ unknowns substituting the results into conditions (5.1), (5.3). We obtain a system T comprising

$$t=m_1-m_2+n_2$$

inequalities relating

$$s=m-m_2+n_2-n$$

unknowns. Optimality of program X is equivalent to solvability of this system of inequalities. The rank of system T is, obviously, s. If the system of inequalities T is solvable, the set of its solutions forms a polyhedral set with vertices.

Hence, to establish solvability of T it suffices to find solutions of all systems of equations comprising s linearly independent restraints entering T. The number of such systems obviously does not exceed C_t^s. If at least one of the solutions satisfies the other relationships of T, the given system of inequalities is solvable and, consequently, X is an optimal program. Otherwise the system of inequalities T has no solutions, which proves that X is not an optimal program.

When s and C_t^s are quite small, the method outlined for establishing the solvability of system T is quite practicable. If, however, the value of these parameters is large the method involves many computations. As a result, an improved version of the method is used in linear programming:

(a) the transition from one system of s equations in s unknowns to another system is ordered, so that the analysis of all the systems becomes superfluous;

(b) each transition is performed with the aid of simple recurrence formulas.

Summing up, we reach the following conclusions:

1. Practical application of the optimality criterion to nondegenerate support programs reduces to solving one system of linear equations.

2. In the degenerate case, the optimality criterion involves analysis of a system of inequalities, which is equivalent to the solution of several systems of linear equations.

5-3. It follows from Theorem 5.2 that the decision vector may be related by conditions (5.4), (5.5) only with the optimal program (4.1)–(4.3). On the other hand, to each optimal program X of this problem a characteristic decision vector related to it by (5.4), (5.5) exists. The question arises as to whether some relationship exists between the decision vector of any optimal program of problem (4.1)–(4.3) and an arbitrary solution of this problem. The following proposition answers this question.

*Theorem 5.3. (**Decision-vector theorem.**) The decision vector of some optimal program of problem (4.1)–(4.3) is also a decision vector for any other solution of this problem.*

Proof. Let $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$ be the decision vector of program X, i. e., a decision vector of problem (4.1)–(4.3) associated with X by (5.4), (5.5). According to Theorem 5.1, the vector Λ solves the dual problem (4.4)–(4.6). In the proof of the second part of Theorem 5.1, we showed that any solution of the dual problem is related by (5.4), (5.5) with any solution of the primal problem. Hence Λ is a decision vector for any optimal program of problem (4.1)–(4.3). This completes the proof.

The above theorem establishes a relationship between the concept of a decision vector and the set of all optimal programs of the given problem. We may, therefore, speak of a decision vector of a linear-programming problem without referring this vector to any particular program.

We emphasize again that in general finding a decision vector of the problem (the decision multipliers) is in no way simpler than solving the problem. However, as stated above, finding a decision vector is equivalent to solving the dual problem. The principal importance of decision multipliers lies in the fact that they are most convenient when stating optimality criteria widely used in computational methods and theoretical applications of linear programming.

In describing linear programming methods we shall generally deal with the canonical form of problems. It is, therefore, expedient to formulate an optimality criterion for this class of linear-programming problems.

Consider problem (1.1)–(1.3) which is obviously a particular case of problem (4.1)–(4.3) with $m_1=0, n_1=n$. The definition of a decision vector for this problem is as follows.

A vector $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$ is called a d e c i s i o n v e c t o r of problem (1.1)–(1.3) if

(a) $$\sum_{i=1}^{m} a_{ij}\lambda_i \geqslant c_j, \quad j=1, 2, \ldots, n;$$

(b) for some program $X=(x_1, x_2, \ldots, x_n)$ of problem (1.1)–(1.3)

$$\sum_{i=1}^{m} a_{ij}\lambda_i = c_j, \quad \text{if} \quad x_j > 0. \tag{5.16}$$

*Theorem 5.4. (**Optimality criterion for problem (1.1)–(1.3).**) A feasible program X of problem (1.1)–(1.3) is optimal if and only if a decision vector exists, related to X by (5.16).*

This theorem is a particular case of the general optimality criterion

(Theorem 5.2). In practical applications the remarks given in 5-2 should be taken into consideration.

5-4. The method of Lagrange multipliers is usually used in solving conditional extremum problems in classical analysis. We review briefly the essential points of this method.

Suppose it is necessary to maximize or minimize a function

$$F(X) = F(x_1, x_2, \ldots, x_n), \tag{5.17}$$

whose variables are related by the conditions

$$G_i(X) = G_i(x_1, x_2, \ldots, x_n) = 0, \quad i = 1, 2, \ldots, m \quad (m < n). \tag{5.18}$$

Let the functions $F(X)$ and $G_i(X)$, $i = 1, 2, \ldots, m$, be continuous and have continuous partial derivatives of the first order with respect to all their variables.

We say that the system (5.18) is r e g u l a r a t p o i n t $\bar{X} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ if the determinant of the matrix

$$\left\| \begin{matrix} \frac{\partial G_1(\bar{X})}{\partial x_{i_1}} & \frac{\partial G_1(\bar{X})}{\partial x_{i_2}} & \cdots & \frac{\partial G_1(\bar{X})}{\partial x_{i_m}} \\ \frac{\partial G_2(\bar{X})}{\partial x_{i_1}} & \frac{\partial G_2(\bar{X})}{\partial x_{i_2}} & \cdots & \frac{\partial G_2(\bar{X})}{\partial x_{i_m}} \\ \cdot & \cdot & \cdot & \cdot \\ \frac{\partial G_m(\bar{X})}{\partial x_{i_1}} & \frac{\partial G_m(\bar{X})}{\partial x_{i_2}} & \cdots & \frac{\partial G_m(\bar{X})}{\partial x_{i_m}} \end{matrix} \right\|$$

does not vanish. Here $i_1, i_2, \ldots, i_m$ are any m subscripts of $(1, 2, \ldots, n)$.

Lagrange's method is based on the following:

If a function (5.17) subject to conditions (5.18) attains a maximum or a minimum at point $\bar{X}$ and the system (5.18) is regular at this point, there exist numbers $\lambda_1, \lambda_2, \ldots, \lambda_m$ such that the function

$$F_\Lambda(X) = F(X) + \sum_{i=1}^{m} \lambda_i G_i(X) \tag{5.19}$$

satisfies, at point $\bar{X}$, the necessary conditions of unconditional extremum, i.e.,

$$\frac{\partial F_\Lambda(\bar{X})}{\partial x_i} = 0 \text{ for } i = 1, 2, \ldots, n.$$

$\lambda_1, \lambda_2, \ldots, \lambda_m$ are generally termed L a g r a n g e m u l t i p l i e r s, and the function $F_\Lambda(X)$— L a g r a n g e f u n c t i o n.

Computation of the conditional extremum (5.17), (5.18) thus reduces to determining the unconditional extremum of the Lagrange function (5.19). The Lagrange function with the unknowns λ_i is determined. Then, the system of equations

$$\frac{\partial F_\Lambda(X)}{\partial x_i} = 0, \qquad i = 1, 2, \ldots, n$$

is solved. The solution of this system depends on the unknown parameters λ_i, $i = 1, 2, \ldots, m$, which are determined from (5.18).

The mathematical-programming problem differs from the classical problem of conditional extremum in that inequality restraints appear. Lagrange's method is, therefore, inapplicable in this case. However, after some

modifications the method can be extended to a fairly general class of mathematical-programming problems, too. Here we consider only the case of linear programming.

Let the linear-programming problem (1.1)–(1.3) be given in canonical form. Let

$$F(X)=\sum_{j=1}^{n} c_j x_j;$$

$$G_i(X)=-\sum_{j=1}^{n} a_{ij}x_j+b_i, \qquad i=1, 2, \ldots, m.$$

Theorem 5.5. A feasible program $\overline{X}$ of problem (1.1)–(1.3) is optimal if and only if the Lagrange function

$$F_\Lambda(X)=F(X)+\sum_{i=1}^{m}\lambda_i G_i$$

for some values of the multipliers $\lambda_i (i=1, 2, \ldots, m)$ attains, at point $\overline{X}$, a maximum subject to the condition

$$x_j \geqslant 0, \qquad i=1, 2, \ldots, n.$$

Proof. Necessity. Let $\overline{X}=(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ be an optimal program of problem (1.1)–(1.3). Let the vector $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$ be an arbitrary decision vector of the problem. Let

$$F_\Lambda(X)=\sum_{j=1}^{n} c_j x_j+\sum_{i=1}^{m}\lambda_i\left(-\sum_{j=1}^{n} a_{ij}x_j+b_i\right)=\sum_{j=1}^{n} x_j\left(c_j-\sum_{i=1}^{m}\lambda_i a_{ij}\right)+\sum_{i=1}^{m}\lambda_i b_i.$$

From the definition of the decision vector,

$$c_j-\sum_{i=1}^{m}\lambda_i a_{ij}\leqslant 0, \qquad j=1, 2, \ldots, n, \tag{5.20}$$

where

$$c_j-\sum_{i=1}^{m}\lambda_i a_{ij}=0, \tag{5.21}$$

if $\bar{x}_j>0$. Applying (5.21), we rewrite the expression for $F_\Lambda(X)$ in the form

$$F_\Lambda(X)=\sum_{j\in E} x_j\left(c_j-\sum_{i=1}^{m}\lambda_i a_{ij}\right)+\sum_{i=1}^{m}\lambda_i b_i,$$

where E is the set of subscripts j for which $\bar{x}_j=0$. Therefore,

$$F_\Lambda(\overline{X})-F_\Lambda(X)=\sum_{j\in E}(\bar{x}_j-x_j)\left(c_j-\sum_{i=1}^{m} a_{ij}\lambda_i\right)=-\sum_{j\in E} x_j\left(c_j-\sum_{i=1}^{m} a_{ij}\lambda_i\right).$$

By assumption, $x_j\geqslant 0$ and the multipliers λ_i satisfy inequalities (5.20). Hence,

$$F_\Lambda(\overline{X})-F_\Lambda(X)\geqslant 0$$

for any vectors X with nonnegative components.

Sufficiency. Let a feasible program $\overline{X}$ of problem (1.1)–(1.3) satisfy the condition

$$F_\Lambda(\overline{X})=\max_{X\geqslant 0} F(X) \tag{5.22}$$

for some $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$. Consider

$$F_\Lambda(\overline{X})=\sum_{j=1}^{n}\bar{x}_j\left(c_j-\sum_{i=1}^{m} a_{ij}\lambda_i\right)+\sum_{i=1}^{m}\lambda_i b_i.$$

If for some j

$$c_j - \sum_{i=1}^{m} a_{ij}\lambda_i > 0,$$

then, letting the component $\bar{x}_j$ of vector $\bar{X}$ increase to infinity we obtain

$$\sup_{X \geqslant 0} F(X) = \infty,$$

which contradicts (5.22). Hence

$$c_j - \sum_{i=1}^{m} a_{ij}\lambda_i \leqslant 0 \quad \text{for} \quad j = 1, 2, \ldots, n. \tag{5.23}$$

Analogous considerations show that

$$c_j - \sum_{i=1}^{m} a_{ij}\lambda_i = 0 \quad \text{for} \quad \bar{x}_j > 0. \tag{5.24}$$

Indeed, if this relationship does not hold, then somewhat modifying the j-th component of the vector $\bar{X}$ we may form a vector X such that

$$F_\Lambda(X) > F_\Lambda(\bar{X}),$$

which contradicts (5.22).

According to Theorem 5.4, relationships (5.23) and (5.24) prove the optimality of program $\bar{X}$. The vector Λ appearing in the function F_Λ is a decision vector of the problem. This completes the proof.

The theorem proved above enables us to refer to the components of the vector $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ appearing in the definition of the function F_Λ for problem (1.1)–(1.3) as the Lagrange multipliers of this problem. In the proof of Theorem 5.5 we established, in particular, that the set of all decision vectors of the linear-programming problem coincides with the system of vectors constructed on the Lagrange multipliers of the problem. Decision multipliers and Lagrange multipliers of a linear-programming problem are thus equivalent concepts.

In our formulation of Theorem 5.5 the vector $\bar{X}$ was assumed to be a feasible program of the problem in question. This theorem, therefore, does not entirely eliminate the necessity of taking into consideration restraints (1.2) relating the variables of problem (1.1)–(1.3). In order to eliminate (1.2), we must find a saddle point of the Lagrange function.

We now give the necessary definitions.

Let $R(X, Y)$ be a function of a vector X, which belongs to the set T_X, and a vector Y, of the set T_Y.

A point $(X_0, Y_0) \in T_X \times T_Y$* is called a saddle point of the function $R(X, Y)$ subject to the condition $(X, Y) \in T_X \times T_Y$, if

$$R(X, Y_0) \leqslant R(X_0, Y_0) \leqslant R(X_0, Y) \tag{5.25}$$

for all the points $(X, Y) \in T_X \times T_Y$. Inequalities (5.25) show that the maximum value of the function $R(X, Y_0)$ on the set T_X is obtained at the point X_0, and the minimum value of the function $R(X_0, Y)$ on the set T_Y at the point Y_0.

We now consider the linear-programming problem (4.1)–(4.3). Restraints (4.2) comprise equalities and inequalities. The nonnegativity requirement is imposed only on part of the variables. Let as before

$$F_\Lambda(X) = F(X, \Lambda) = \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{m} \lambda_i \Big(b_i - \sum_{j=1}^{n} a_{ij} x_j\Big) = \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{m} \lambda_i b_i - \sum_{i=1}^{m} \sum_{j=1}^{n} \lambda_i x_j a_{ij}.$$

* By definition $W = (u, v) \in T_X \times T_Y$, if $u \in T_X$, $v \in T_Y$.

Theorem 5.6. The vectors $X^* = (x_1^*, x_2^*, \ldots, x_n^*)$ *and* $\Lambda^* = (\lambda_1^*, \lambda_2^*, \ldots, \lambda_m^*)$ *are, respectively, a solution of problem (4.1)-(4.3) and its decision vector if and only if* (X^*, Λ^*) *is a saddle point of the function* $F(X, \Lambda)$ *subject to the conditions*

$$\left.\begin{array}{l} x_j \geqslant 0, \quad j = 1, 2, \ldots, n_1, \\ \lambda_i \geqslant 0, \quad i = 1, 2, \ldots, m_1. \end{array}\right\} \tag{5.26}$$

Proof. The proof is based on arguments very similar to those used in the proof of the Theorem 5.5. The suitable changes are left to the reader (see Exercise 13).

According to Theorem 5.6, a pair of conjugate problems (4.1)-(4.3) and (4.4)-(4.6) is equivalent to the problem of finding a saddle point of the Lagrange function $F(X, \Lambda)$ subject to the conditions (5.26).

Theorem 5.6, formulated here for linear-programming problems, can be extended to a wide class of nonlinear-programming problems.

This generalization of Theorem 5.6 provides a theoretical basis for some numerical methods of nonlinear programming.

5-5. To conclude this section, we give another interpretation of decision multipliers of linear-programming problems. We limit the discussion to problem (1.1)-(1.3) written in canonical form.

Let $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ be a decision vector of problem (1.1)-(1.3) or, equivalently, a solution of the dual problem (1.4)-(1.5). We shall now show that the components λ_i of vector Λ may be interpreted as the influence of the various restraints of system (1.2) on the magnitude of the maximum in problem (1.1)-(1.3). We state this proposition more precisely.

Theorem 5.7. Let problem (1.1)-(1.3) be nondegenerate and let $M(b_1, b_2, \ldots, b_m)$ *be the maximum of its linear form under conditions (1.2), (1.3). Then*

$$\lambda_i = \frac{\partial M(b_1, b_2, \ldots, b_m)}{\partial b_i}, \quad j = 1, 2, \ldots, m. \tag{5.27}$$

Proof. We denote problem (1.1)-(1.3) by (A_B). Let $X^* = (x_1^*, x_2^*, \ldots, x_n^*)$ be a support solution of (A_B). Without loss of generality we may take the first components of the vector m to be nonzero. The restraint vectors A_j, $j = 1, 2, \ldots, m$ are then linearly independent and any m-dimensional vector $B' = (b_1', b_2', \ldots, b_m')$ can be written as a linear combination of the restraint vectors

$$B' = \sum_{i=1}^{m} x_i' A_i. \tag{5.28}$$

Let $\|e_{ij}\|_m$ denote the inverse of the regular matrix $(A_1, A_2, \ldots, A_m)$. Then, from (5.28), we have

$$x_i' = \sum_{j=1}^{m} e_{ij} b_j', \qquad i = 1, 2, \ldots, m. \tag{5.29}$$

Let

$$e = \max_{1 \leqslant i \leqslant m} \sum_{j=1}^{m} |e_{ij}|; \quad x = \min_{1 \leqslant i \leqslant m} x_i^*. \tag{5.30}$$

Consider a linear-programming problem $(A_{B'})$ which is obtained from problem (A_B) by replacing B by B'. We shall show that when

$$\max_{1 \leqslant i \leqslant m} |b_i - b_i'| \leqslant \frac{x}{e} \tag{5.31}$$

the n-dimensional vector $X'=(x'_1, x'_2, \ldots, x'_m, 0, 0, \ldots, 0)$, where the components x'_i are defined in (5.29), is a solution of problem $(A_{B'})$.

We shall first show that X' is a feasible program of problem $(A_{B'})$. According to (5.29),

$$x'_i - x^*_i = \sum_{j=1}^{m} e_{ij}(b'_j - b_j), \qquad i = 1, 2, \ldots, m.$$

Hence, applying notations (5.30) and condition (5.31),

$$|x'_i - x^*_i| \leqslant e \max |b'_j - b_j| \leqslant x,$$

and, therefore,

$$x'_i \geqslant x^*_i - x \geqslant 0.$$

The vector X' thus satisfies restraints (1.3) of $(A_{B'})$. The restraints (1.2), with B' substituted for B, are satisfied by X' because of the particular definition of the components $x'_1, x'_2, \ldots, x'_m$ (see (5.28)). X' is thus a feasible program of $(A_{B'})$.

Let $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ be an optimal program of problem (1.4)-(1.5) (a decision vector of (A_B)). According to the optimality criterion of programs of (A_B) (see Theorem 5.4), we have

$$\sum_{i=1}^{m} a_{ij}\lambda_i \begin{cases} = c_j, & j = 1, 2, \ldots, m, \\ \geqslant c_j, & j = m+1, m+2, \ldots, n. \end{cases} \tag{5.32}$$

Relationships (5.32) show that Λ is also a decision vector of $A_{B'}$ related by (5.16) with the feasible program X'. Vector X' is thus a solution of $A_{B'}$.

When conditions (5.31) are satisfied, X' is a solution of $A_{B'}$, and Λ (independent of the choice of B') is a solution of the dual problem with respect to $A_{B'}$.

According to the first duality theorem

$$M(b'_1, b'_2, \ldots, b'_m) = \sum_{i=1}^{m} b'_i\lambda_i, \tag{5.33}$$

where B' is restrained only by (5.31).

Since

$$x = \min_{1 \leqslant i \leqslant m} x^*_i$$

is positive (the program X^* is, by assumption, nondegenerate), we conclude that (5.33) holds for any vector B' lying in some given neighborhood of vector B. The required equalities (5.27) can now be obtained by differentiation of (5.33). This completes the proof.

Thus, in the nondegenerate case the components of a decision vector of problem (1.1)-(1.3) give an appraisal of the influence of the right-hand sides of restraints (1.2) on the magnitude of the attainable maximum of linear form (1.1).

In the proof of Theorem 5.7 we have not fully drawn on the assertion that all the support programs of a problem are nondegenerate. It was only assumed that one of the optimal programs is nondegenerate. It is noteworthy that the last requirement is essential. If this condition is not fulfilled, the solution of the dual problem is generally not unique (see 6-2). Formula (5.27) will, therefore, not hold in general: the function $M(b_1, \ldots, b_m)$ need not have partial derivatives. However, in the degenerate case, too, our interpretation of the decision multipliers remains valid (though in a somewhat modified form).

Formula (5.27) was derived for the linear-programming problem in canonical form. Obviously, however, this formula applies to any problem with a nondegenerate support solution. We suggest that the reader prove, as an exercise, Theorem 5.7 for problem (4.1)–(4.3) (see Exercise 14).

§ 6. Some applications of the duality principle

The duality of conjugate linear-programming problems established in this chapter is most useful in linear programming and in related mathematical disciplines. The duality principle is used in the construction of various numerical methods of linear programming. This aspect is dealt with in Chapters 6 and 7. Moreover, the concept of duality is widely applied in qualitative investigations of various mathematical problems. The principle of duality becomes particularly effective when extended to infinite-dimensional spaces. However, even in the finite-dimensional case, which is considered here, duality simplifies the analysis of certain mathematical problems. We shall illustrate this point by several examples.

6-1. Consider the linear-programming problem of maximizing the linear form

$$\sum_{i=1}^{m} c_i y_i \tag{6.1}$$

subject to the conditions

$$\sum_{i=1}^{m} d_{ij} y_i \leqslant d_j, \qquad j=1, 2, \ldots, n. \tag{6.2}$$

We shall assume that problem (6.1)–(6.2) is solvable. In this case, all its solutions (the optimal set of the problem) is a convex polyhedral set M^*. We shall determine the dimensionality of the optimal set M^* and, in particular, establish the conditions under which this set is zero-dimensional (condition under which a unique solution of the problem exists). To this end we formulate the dual problem of problem (6.1)–(6.2).

Minimize the linear form

$$\sum_{j=1}^{n} d_j x_j \tag{6.3}$$

subject to the conditions

$$\sum_{j=1}^{n} d_{ij} x_j = c_i, \qquad i=1, 2, \ldots, m; \tag{6.4}$$

$$x_j \geqslant 0, \; j = 1, 2, \ldots, n. \tag{6.5}$$

We call the restraint vector $D_j=(d_{1j}, d_{2j}, \ldots, d_{mj})^T$ of problem (6.3)–(6.5) free if the j-th restraint in (6.5) is free (see 3-4).

Theorem 6.1. ***The optimal set M^* of problem (6.1)–(6.2) is ϱ^*-dimensional, where***

$$\varrho^* = m - r, \tag{6.6}$$

where r is the rank of the matrix of the free restraint vectors of problem (6.3)–(6.5).

Proof. Let E be the set of indices of the free vectors of problem (6.3)–(6.5). In this case, according to the second duality theorem, the j-th restraint in (6.2) for $j \in E$ is fixed, i.e., for any vector $Y \in M^*$

$$\sum_{i=1}^{m} d_{ij} y_i = d_j.$$

Consequently, the optimal set M^* is contained in the polyhedral set M' defined by

$$\sum_{i=1}^{m} d_{ij} y_i \begin{cases} \leqslant d_j, & j \notin E, \quad j=1, 2, \ldots, n; \\ = d_j, & j \in E, \quad j=1, 2, \ldots, n. \end{cases} \tag{6.7, 6.8}$$

We shall show that any vector Y of M' is a solution of problem (6.1)–(6.2), i.e., belongs to M^*.

Let $X^* = (x_1^*, x_2^*, \ldots, x_n^*)$ be some solution of problem (6.3)–(6.5). From the definition of the set E,

$$x_j^* = 0 \text{ for } j \notin E. \tag{6.9}$$

Consider an arbitrary vector $Y \in M'$. The following equalities hold:

$$\sum_{i=1}^{m} c_i y_i = \sum_{i=1}^{m} \left(\sum_{j \in E} d_{ij} x_j^* \right) y_i =$$
$$= \sum_{j \in E} x_j^* \sum_{i=1}^{m} d_{ij} y_i =$$
$$= \sum_{j \in E} x_j^* d_j =$$
$$= \sum_{j=1}^{n} x_j^* d_j.$$

The first equality follows from (6.9) and (6.4), the third and fourth follow from (6.8) and (6.9), respectively. The second equality holds when the order of summation is reversed.

Thus,

$$\sum_{i=1}^{m} c_i y_i = \sum_{j=1}^{n} d_j x_j^*. \tag{6.10}$$

According to Lemma 1.2, (6.10) indicates that program Y is a solution of problem (6.1)–(6.2).

M^* thus coincides with M' and is, consequently, defined by conditions (6.7), (6.8).

From the definition of E, any j-th restraint of system (6.5) when $j \notin E$ is fixed. Hence, according to the second duality theorem, the j-th restraint in (6.7) ($j \notin E$) is free. In other words, for any $j \notin E$ there exists a vector $Y^{(j)} \in M^*$ so that

$$\sum_{i=1}^{m} d_{ij} y_i^{(j)} < d_j.$$

All the conditions of system (6.7) thus impose nonrigid constraints on the polyhedral set M^* (see Chapter 2, 1-2).

It now remains to apply Theorem 1.2, Chapter 2, and (6.6) follows. This completes the proof.

As a corollary of Theorem 6.1, we derive the necessary and sufficient conditions for the existence of a unique solution of problem (6.1)–(6.2).

Theorem 6.2. *The solution of problem (6.1)-(6.2) is unique if and only if there are linearly independent restraint vectors among those of problem (6.3)-(6.5).*

The proof follows immediately from (6.6) for $r = m$.

We now give one sufficient condition for the existence of a unique solution of problem (6.1)–(6.3).

Theorem 6.3. *If at least one of the optimal support programs of problem (6.3)-(6.5) is nondegenerate, problem (6.1)-(6.2) has a unique solution.*

The proof follows directly from the sufficiency of the conditions of Theorem 6.2.

If we assume that problem (6.3)–(6.5) has a unique solution, the conditions of Theorem 6.3 prove to be not only sufficient but also necessary. Indeed, let $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ be the unique solution of problem (6.3)–(6.5). In this case the restraint vector D_j is free if and only if $x_j^* > 0$. If the support program X^* is degenerate, the number of $x_j^* > 0$ is less than m. Hence, from Theorem 6.1, the dimensionality of the optimal set M^* is greater than zero, which means that (6.1)–(6.2) has not a unique solution.

If problem (6.3)–(6.5) has many solutions, the conditions of Theorem 6.3, generally speaking, are only sufficient. This can be verified by a suitable example, the construction of which is left to the reader (Exercise 15).

6-2. Until now we studied problem (6.1)–(6.2) with fixed parameters c_i, d_{ij} and d_j. We now slightly modify the statement of our object and consider all solvable problems (6.1)–(6.2) with equal c_i, d_{ij} and arbitrary d_j. The set of linear-programming problems which is obtained in this way will be denoted by $H(c_i, d_{ij})$. We are interested in finding the necessary and sufficient conditions for a unique solution of all problems of this class.

Theorem 6.4. *Nondegeneracy of problem (6.3)-(6.5) is a necessary and sufficient condition for each of the problems of $H(c_i, d_{ij})$ to have a unique solution.*

Proof. Sufficiency is an obvious corollary of Theorem 6.3. We will prove necessity.

Let problem (6.3)–(6.5) be degenerate. This means that it has at least one degenerate support program

$$X^* = (x_1^*, x_2^*, \dots, x_n^*).$$

To be specific, let

$$x_j^* \begin{cases} > 0, & j = 1, 2, \dots, r < m, \\ = 0, & j = r+1, \dots, n. \end{cases}$$

Let $\overline{Y}=(\overline{y}_1, \overline{y}_2, \ldots, \overline{y}_m)$ be a solution of the system of equations

$$\sum_{i=1}^{m} d_{ij}y_i=\overline{d}_j, \qquad j=1, 2, \ldots, r, \tag{6.11}$$

where $\overline{d}_j$, $j=1, 2, \ldots, r$ are arbitrary numbers. For $\overline{d}_j$ when $j>r$ we take any d_j satisfying the condition

$$\sum_{i=1}^{m} d_{ij}\overline{y}_i<d_j, \qquad j=r+1, r+2, \ldots, n. \tag{6.12}$$

We shall show that a problem of $H(c_i, d_{ij})$ for $d_j=\overline{d}_j$ has several solutions.

Let $Y^{(0)}=(y_1^{(0)}, y_2^{(0)}, \ldots, y_m^{(0)})\neq 0$ be a solution of the homogeneous system of equations corresponding to system (6.11). Existence of this vector follows from the fact that $r<m$. We form a system G comprising the vectors $Y_\varepsilon=\overline{Y}+\varepsilon Y^{(0)}$, where the parameter ε is chosen so that the vector Y_ε is a feasible program of problem (6.1)-(6.2). Obviously, there exists a number $\alpha>0$ such that Y_ε for $|\varepsilon|\leqslant\alpha$ satisfies (6.2). The system G therefore comprises an infinity of vectors, each of which is a solution of system (6.11). Let Y be any vector of G. Following the reasoning used in the proof of (6.10), we obtain

$$\sum_{i=1}^{m} c_i y_i=\sum_{j=1}^{n} \overline{d}_j x_j^*.$$

The vector Y is thus a solution of problem (6.1)-(6.2).

Assuming degeneracy of problem (6.3)-(6.5) we have thus found a problem of $H(c_i, d_{ij})$ with several solutions. Necessity of the conditions of Theorem 6.4 is thus established.

Let

$$\Delta(j_1, j_2, \ldots, j_{m-1})=\begin{vmatrix} d_{1j_1} & d_{1j_2} & \ldots & d_{1j_{m-1}} & c_1 \\ d_{2j_1} & d_{2j_2} & \ldots & d_{2j_{m-1}} & c_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ d_{mj_1} & d_{mj_2} & \ldots & d_{mj_{m-1}} & c_m \end{vmatrix}.$$

As a corollary of Theorem 6.4, we shall establish a sufficient condition for problems of $H(c_i, d_{ij})$ to have a unique solution.

Theorem 6.5. *If for any system of linearly independent restraint vectors $D_{j_1}, D_{j_2}, \ldots, D_{j_{m-1}}$ of problem (6.3)-(6.5) the determinant $\Delta(j_1, j_2, \ldots, j_{m-1})$ does not vanish and the matrix $(D_1, D_2, \ldots, D_n)$ is of rank m, all the problems of $H(c_i, d_{ij})$ have unique solutions.*

Proof. Let the system of vectors $D_{s_1}, D_{s_2}, \ldots, D_{s_m}$ constitute the basis of some support program $X=(x_1, x_2, \ldots, x_n)$ of problem (6.3)-(6.5). In this case $x_j=0$ for $j\neq s_k$, $k=1, 2, \ldots, m$. The components of program X expressed in term of the basis vectors can be determined from the system of equations

$$\sum_{k=1}^{m} d_{is_k}x_{s_k}=c_i, \qquad i=1, 2, \ldots, m.$$

Hence, applying Cramer's rule, we obtain

$$x_{s_k}=\frac{(-1)^{m-k}\Delta(s_1, s_2, \ldots, s_{k-1}, s_{k+1}, \ldots, s_m)}{\Delta}, \quad k=1, 2, \ldots, m,$$

where Δ is the determinant constructed from the components of the vectors $D_{s_1}, D_{s_2}, \ldots, D_{s_m}$. By assumption, however,

$$\Delta(s_1, s_2, \ldots, s_{k-1}, s_{k+1}, \ldots, s_m)\neq 0.$$

Therefore $x_{s_k}\neq 0$ for $k=1, 2, \ldots, m$. The vector X is thus a nondegenerate program of the problem.

Since X is an arbitrary support program of problem (6.3)-(6.5), this problem is nondegenerate. Hence, from Theorem 6.4, each problem of $H(c_i, d_{ij})$ has a unique solution. This completes the proof.

6-3. To give another application of the duality principle, we shall prove two basic theorems of the theory of linear inequalities. Consider a system of linear inequalities

$$\sum_{i=1}^{m} d_{ij}y_i\leqslant d_j, \qquad j=1, 2, \ldots, n. \tag{6.13}$$

Theorem 6.6. ***(Consistency of system (6.13).) The system of inequalities (6.13) is consistent if and only if any nonnegative solution*** $X=(x_1, x_2, \ldots, x_n)$ ***of the homogeneous system of equations***

$$\sum_{j=1}^{n} d_{ij}x_j=0, \qquad i=1, 2, \ldots, m, \tag{6.14}$$

satisfies the condition

$$\sum_{j=1}^{n} d_j x_j \geqslant 0. \tag{6.15}$$

Proof. Necessity is easily established. Indeed, let system (6 13) be consistent and let $Y=(y_1, y_2, \ldots, y_m)$ be one of its solutions Multiplying the j-th inequality in (6.13) by the nonnegative number x_j and summing the products, we obtain

$$\sum_{i=1}^{m} y_i\left(\sum_{j=1}^{n} d_{ij}x_j\right) \leqslant \sum_{j=1}^{n} d_j x_j. \tag{6.16}$$

If the vector $X=(x_1, x_2, \ldots, x_n)$ satisfies equations (6.14), inequality (6 15) follows from (6.16)

We now prove sufficiency. Consider the linear-programming problem (6.1)–(6.2), where $c_i=0$ for $i=1, 2, \ldots, m$. We shall denote this problem by (A). Any solution of system (6.13) is obviously a solution of problem (A) and, conversely, any solution of problem $(\tilde{A})$ satisfies (6.13). The consistency of system (6.13) is therefore equivalent to solvability of problem (A).

Let $(\tilde{A})$ be the dual problem with respect to (A). Problem $(\tilde{A})$ has the form (6.3)–(6.5) for $c_i=0$, $i=1, 2, \ldots, m$. Let the requirements of the thorem be satisfied. In terms of problem $(\tilde{A})$ this means that the linear form of problem $(\tilde{A})$ is bounded below by zero in the set of its feasible programs. Since the set of feasible programs of problem $(\tilde{A})$ is a priori nonempty (one of its programs is the zero vector), problem $(\tilde{A})$ is solvable according to Chapter 2, Theorem 4.4. In this case, however, the first duality theorem guarantees the solvability of problem (A), which is equivalent to consistency of the system of inequalities (6.13). This completes the proof.

Note that this theorem is a natural analog of the well-known consistency criterion for systems of linear equations (see Appendix, Theorem 2.6), which is stated as follows:

A system of linear equations

$$\sum_{i=1}^{m} d_{ij}y_i=d_j, \qquad j=1, 2, \ldots, n,$$

is consistent if and only if any solution

$$X=(x_1, x_2, \ldots, x_n)$$

of the conjugate homogeneous system

$$\sum_{j=1}^{n} d_{ij}x_j=0, \qquad i=1, 2, \ldots, m,$$

satisfies the relationship

$$\sum_{j=1}^{n} d_j x_j=0.$$

Together with system of inequalities (6.13), consider an arbitrary inequality

$$\sum_{i=1}^{m} c_i y_i \leqslant c. \tag{6.17}$$

A question arises: under what conditions does inequality (6.17) follow from (6.13)?

The answer is provided by the following theorem.

Theorem 6.7. Inequality (6.17) follows from the consistent system (6.13) if and only if there exist nonnegative numbers $x_1^*, x_2^*, \ldots, x_n^*$ ***such that***

$$\sum_{j=1}^{n} d_{ij}x_j^*=c_i, \qquad i=1, 2, \ldots, m, \tag{6.18}$$

$$\sum_{j=1}^{n} d_j x_j^* \leqslant c. \tag{6.19}$$

2154

Proof. Sufficiency is almost obvious. Indeed, let the requirements of the theorem be fulfilled and let $Y=(y_1, y_2, \ldots, y_m)$ be a solution of system (6.13). Multiplying the j-th inequality in (6.13) by x_j^* and summing the products, we obtain, applying (6.18),

$$\sum_{i=1}^{m} c_i y_i \leqslant \sum_{j=1}^{n} d_j x_j^*.$$

Further applying inequality (6.19), we conclude that Y satisfies (6.17).

To prove necessity, consider the linear-programming problem (6.1)–(6.2). The value of the linear form (6.1) of any feasible program of problem (6.1)–(6.2) is at most c (any solution of system (6.13) satisfies (6.17)). The set of feasible programs of the given problem is nonempty (inequalities (6.13) are consistent). Therefore, according to Chapter 2, Theorem 4.4, problem (6.1)–(6.2) is solvable.

Let $Y^*=(y_1^*, y_2^*, \ldots, y_m^*)$ be a solution of problem (6.1)–(6.2). Since the vector Y^* satisfies inequalities (6.13),

$$\sum_{i=1}^{m} c_i y_i^* \leqslant c. \qquad (6.20)$$

According to the first duality theorem problem (6.3)–(6.5), dual with respect to the solvable problem (6.1)–(6.2), has a solution $X^*=(x_1^*, x_2^*, \ldots, x_n^*)$ which satisfies the equality

$$\sum_{i=1}^{m} c_i y_i^* = \sum_{j=1}^{n} d_j x_j^*. \qquad (6.21)$$

The vector X^* is a feasible program of problem (6.3)–(6.5). Its components, therefore, satisfy conditions (6.18). As regards inequality (6.19), it is satisfied by virtue of (6.20) and (6.21). This proves necessity.

Like the previous theorem, this proposition is also an analog of a corresponding assertion for linear equations:

The equation

$$\sum_{i=1}^{m} c_i y_i = c$$

follows from the system of equations

$$\sum_{i=1}^{m} d_{ij} y_i = d_j, \qquad j=1, 2, \ldots, n,$$

if and only if there exists $x_1^*, x_2^*, \ldots, x_n^*$ *such that*

$$\sum_{j=1}^{n} x_j^* d_{ij} = c_i, \qquad i=1, 2, \ldots, m;$$

$$\sum_{j=1}^{n} x_j^* d_j = c.$$

EXERCISES TO CHAPTER 3

1. Formulate the dual problem with respect to the transportation problem (see Chapter 1, 3-2).

2. Formulate the dual problem with respect to the problem of national-economy planning discussed in Chapter 1, 7-3. Give the economic interpretation of the dual problem.

3. Let the vectors $T_i \neq 0$, $i=1, 2, \ldots, N$; and let T_P be a convex polyhedral cone spanned by the vectors T_i and having its apex at . For apex P to be the point of cone T_P it is necessary and sufficient that:

For any $y_i \geqslant 0$,

$$\sum_{i=1}^{N} y_i T_i = 0$$

entails

$$y_1 = y_2 = \cdots = y_N = 0.$$

Prove.

4. Prove that the set $T_P^{(0)}$ introduced in the proof of Lemma 2.3 is a convex polyhedron (see Chapter 2, Exercise 13).

5. Prove that Lemma 2.3 applies if T_P is an arbitrary convex cone with its point at P.

6. Give examples showing that if the assumptions of Lemma 2.3 concerning the cone K or the point P are changed, the lemma, generally speaking, does not hold.

7. Show that there does not exist a pair of dual linear-programming problems whose linear forms are unbounded in the corresponding sets of feasible programs.

8. Show that the problem with homogeneous restraints (1.17)–(1.19) is solvable if the components of its constraint vector are nonnegative and one of the following conditions is satisfied:

(a) for some i, all $a_{ij} > 0$.

(b) $a_{ij} \geqslant 0$ for all i and j, and for any j there exists at least one positive a_{ij}.

9. Prove the first duality theorem for maximization of the linear form

$$L(X) = \sum_{j=1}^{n} c_j x_j \tag{1}$$

subject to the conditions

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \qquad i = 1, 2, \ldots, N, \tag{2}$$

by showing that if X^* is a solution of problem (1)–(2), the point

$$\bar{C} = (c_1, c_2, \ldots, c_n, L(X^*))$$

belongs to the convex polyhedral cone K spanned by the vectors

$$\bar{A}^{(i)} = (a_{i1}, a_{i2}, \ldots, a_{in}, b_i), \qquad i = 1, 2, \ldots, N,$$

and

$$e_{n+1} = (0, 0, \ldots, 0, 1).$$

Hint: Prove the last proposition by "reductio ad absurdum", applying Corollary 3.2, Chapter 2, to point $\bar{C}$ and to cone K.

10. Verify that problem (4.1)–(4.3) is dual with respect to problem (4.4)–(4.6) (see 4-1).

11. Use Theorem 5.2 to derive an optimality criterion for programs of problems indicated in Exercise 1.

12. Test, for optimality, the programs (1, 0, 0, 0) and (0, 0, 1, 1) of the problem of maximization of the linear form

$$x_1 - 3x_2 + x_4$$

subject to the conditions

$$\begin{gathered} x_1 - x_2 + 2x_4 = 1, \\ 2x_1 + x_2 + 4x_3 - 2x_4 = 2, \\ x_1 \geqslant 0, \quad x_2 \geqslant 0, \; x_3 \geqslant 0, \quad x_4 \geqslant 0. \end{gathered}$$

13. Prove Theorem 5.6.

14. Prove the validity of (5.27) for problem (4.1)–(4.3) with at least one nondegenerate support solution.

15. Show that there exists a unique solution of problem (6.1)–(6.2) although its dual problem has no nondegenerate support solutions.

Chapter 4

THEORETICAL PRINCIPLES OF THE SIMPLEX METHOD

The simplex method is the method most widely used in applications of linear programming. The foundations of this method were laid by Dantzig in 1949 /40, 41/. In Soviet literature on linear programming the Dantzig method is sometimes referred to as the method of successive program improvement. The term simplex method arose from the geometrical interpretation of the first particular problems to which it was applied* and, although universally accepted, it does not adequately describe the essential features of the procedure.

Conceptually, the method entails three main points. First, a procedure for computing a support program is indicated. Second, a criterion is established for testing the optimality of the program selected (or, in other words, for deciding whether the program selected is a solution). Third, a procedure is proposed for transforming a nonoptimal program selected to another support program which is closer to the optimum. It is proved that the procedure yields an optimal program, i. e., a solution of the linear-programming problem, after a finite number of steps. Thus, the method is a successive improvement of program, which is expressed in the alternative name, mentioned above, given to the method.

It is noteworthy that the algorithms of the simplex method also make it possible to establish, during computations, whether the linear-programming problem in question is solvable. In other words, when computing it is possible to establish whether the problem restraints are contradictory and whether they ensure boundedness of the linear form.

In this chapter we discuss only the theoretical principles of the method. The computational procedures are dealt with in the next chapter.

The simplex method is presented as follows. First we give two optimality tests for a support program of the linear-programming problem. Then we discuss the method in reference to the so-called nondegenerate case. Some support program of the problem in question is assumed to be given. The formal exposition is supplemented by two geometrical interpretations.

In § 5 the simplex method is generalized and discussed for linear-programming problems with upper, as well as lower, bounds. Thereafter the limitation of nondegeneracy is dropped.

* Restraints of the form

$$\sum_{j=1}^{n} x_j = 1, \quad x_j \geqslant 0, \quad j = 1, 2, \ldots, n$$

define a simplex in the n-dimensional space. These restraints comprised the conditions for one of the first linear-programming problems for which Dantzig developed a computational procedure.

In § 7 we describe the procedure for constructing an initial support program and the so-called M-method. At the end of the chapter we give some theoretical applications of the simplex method.

§ 1. Optimality tests for support programs

1-1. In Chapter 3 we derived an optimality criterion, i. e., a necessary and sufficient condition for a feasible program to be a solution of the linear-programming problem. In the simplex method we do not deal with any feasible program, but only with support programs. Thus, in this section we shall formulate an optimality test for a support program. The optimality test can be expressed in terms of the various characteristic parameters of the problem. We shall deal with two different optimality tests. Each of these tests will be used in one of the solution algorithms of the linear-programming problem according to the simplex method. Note that we speak here of optimality *tests* (rather than *criteria*) to emphasize that the considerations apply in particular to support programs and that the conditions derived are, generally speaking, not sufficient for any feasible program.

Let us write the linear-programming problem in canonical form.

Maximize the linear form

$$L(X)=\sum_{j=1}^{n}c_jx_j \tag{1.1}$$

subject to the conditions

$$\sum_{j=1}^{n}A_jx_j=B; \tag{1.2}$$

$$x_j\geqslant 0,\quad j=1,\ 2,\ \dots,\ n. \tag{1.3}$$

The problem can also be written more compactly: Compute a nonnegative vector (row vector) $X\geqslant 0$ maximizing $L(X)=CX^T$ subject to the conditions $AX^T=B$.

Here $A=(A_1,\ \dots,\ A_n)=\|a_{ij}\|_{mn}$ is the restraint matrix, A_j and B being, respectively, the restraint vector and the constraint vector of the problem: $A_j=(a_{1j},\ a_{2j},\ \dots,\ a_{mj})^T, j=1,\ 2,\ \dots, n$; $B=(b_1,\ b_2,\ \dots,\ b_m)^T$; $C=(c_1,\ c_2,\ \dots, c_n)$ is the given coefficient vector of the linear form (1.1). We shall assume that the matrix A is of rank m.

We restate the definition of a support program and its basis, these being the principal concepts of the present chapter. A feasible program $X=(x_1,\ x_2,\ \dots, x_n)$ of the linear-programming problem (given in canonical form (1.1)–(1.3)) is called a *support program* if the system of the restraint vectors A_j corresponding to its positive components $(x_j>0)$ is linearly independent. The *basis* of the support program is the system of m linearly independent restraint vectors which contains *all* the A_j corresponding to the positive components of the support program. *Basis* components of the support program are the components corresponding to its basis vector; the remainder are called *extrabasis* variables.

Let some support program X with the basis $A_{s_1},\ A_{s_2},\ \dots,\ A_{s_i},\ \dots,\ A_{s_m}$ be given. In the following we shall find it convenient to describe any basis vector not only by the index s_i but also by its position i in the particular basis (for any basis, obviously, $i=1,\ 2,\ \dots,\ m$).

Let us take some restraint vector A_j and express it in terms of the basis vectors $A_{s_1}, \ldots A_{s_m}$:

$$A_j = \sum_{i=1}^{m} A_{s_i} x_{ij}, \qquad j = 1, 2, \ldots, n. \tag{1.4}$$

Here x_{ij} is the coefficient in the expansion of the vector A_j in terms of the basis vectors corresponding to the vector A_{s_i} which occupies the i-th position in the basis.

Consider the set of indices I_X of the basis vectors of the support program X (I_X is the set of subscripts of the basis variables). Obviously

$$B = \sum_{i=1}^{n} x_j A_j = \sum_{j \in I_X} x_j A_j$$

(the extrabasis variables are all zero). The constraint vector B as expressed in terms of basis vectors is

$$B = \sum_{i=1}^{m} A_{s_i} x_{i0},$$

where $x_{i0} = x_{s_i}$ is the basis variable corresponding to the vector A_{s_i} which occupies the i-th position in the basis.

Denoting the constraint vector B by A_0, we obtain a general formula

$$A_j = \sum_{i=1}^{m} A_{s_i} x_{ij}, \qquad j = 0, 1, 2, \ldots, m, \tag{1.4'}$$

defining the coefficients of the expansion of all the restraint vectors and of the constraint vector in terms of basis vectors.

Consider the sets of the parameters z_j and $\Delta_j (j = 1, 2, \ldots, n)$ defined as follows:

$$z_j = \sum_{i=1}^{m} c_{s_i} x_{ij}, \qquad j = 1, 2, \ldots, n; \tag{1.5}$$

$$\Delta_j = z_j - c_j, \qquad j = 1, 2, \ldots, n. \tag{1.6}$$

The parameters z_j and Δ_j are determined by the support program X corresponding to the basis $A_{s_1}, \ldots, A_{s_m}$. In order to emphasize this we should write $z_j^{(X)}$ and $\Delta_j^{(X)}$. However, to avoid unnecessary complication of notations, we shall omit the superscript X whenever no confusion can arise.

The parameters Δ_j are most significant in the simplex method: the signs of these parameters indicate the optimality of the program chosen. We have:

Optimality test. A support program X^* is a solution of problem (1.1)-(1.3) if $\Delta_j \geqslant 0$ for $j = 1, 2, \ldots, n$.

Proof. Consider a feasible program $X = (x_1, \ldots, x_n) \neq X^*$. The components of the vector X, like those of any feasible program, should satisfy the restraints of the problem, i. e.,

$$B = \sum_{j=1}^{n} x_j A_j,$$

$$x_j \geqslant 0, \qquad j = 1, 2, \ldots, n.$$

Expressing the vector A_j, according to (1.4), in terms of the basis vectors of the support program X^* we obtain

$$B = \sum_{j=1}^{n} x_j \left(\sum_{i=1}^{m} x_{ij} A_{s_i}\right),$$

or, reversing the order of summation,

$$B=\sum_{i=1}^{m}\left(\sum_{j=1}^{n}x_jx_{ij}\right)A_{s_i}. \tag{1.7}$$

On the other hand, X^* being a feasible program of problem (1.1)-(1.3), we have

$$B=\sum_{i=1}^{m}x_{i0}^*A_{s_i}. \tag{1.8}$$

Since A_{s_1}, A_{s_2}, ..., A_{s_m} are linearly independent, we can express the constraint vector B as a unique linear combination of the basis vectors. Comparing (1.7) with (1.8), we see that

$$\sum_{j=1}^{n}x_jx_{ij}=x_{i0}^*. \tag{1.9}$$

The validity of our proposition now follows from the following equalities and inequalities:

$$\begin{aligned} L(X)&=\sum_{j=1}^{n}c_jx_j\leqslant \\ &\leqslant\sum_{j=1}^{n}z_jx_j= \\ &=\sum_{j=1}^{n}\left(\sum_{i=1}^{m}c_{s_i}x_{ij}\right)x_j= \\ &=\sum_{i=1}^{m}\left(\sum_{j=1}^{n}x_jx_{ij}\right)c_{s_i}= \\ &=\sum_{i=1}^{m}c_{s_i}x_{i0}^*= \\ &=L(X^*). \end{aligned}$$

The first and the last equalities give the value of the linear form L on programs X and X^*, respectively. The inequality follows from the conditions $\Delta_j=z_j-c_j\geqslant 0$. The third and fifth equalities follow from (1.5) and (1.9), respectively, and the fourth equality is obtained by reversing the order of summation.

Thus, if $\Delta_j\geqslant 0$ for all j, the support program X determines the maximum value of the linear form $L(X)$ and as such solves the linear-programming problem. We emphasize that this condition ($\Delta_j\geqslant 0$ for all j) is, in general, only a sufficient condition for the optimality of a support program (see Exercise 1).

As we have already noted, Δ_j can be expressed in terms of the various parameters characteristic of the linear programming problem. Depending on the definition of z_j in (1.6), we have different optimality tests.

We shall say that the optimality test is given in the first form if z_j are computed from (1.5). Thus the first form of the optimality test for support programs is formulated as follows:

Optimality test. A support program X^* is a solution of problem (1.1)-(1.3) if

$$\sum_{i=1}^{m}c_{s_i}x_{ij}\geqslant c_j$$

for all j ($j=1, 2, \ldots, n$).

1-2. We now give an alternative form of the optimality test for support programs.

Let, as before, the vectors $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$ constitute a basis of the support program X^*. We determine the vector $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$ from the system of equations

$$\sum_{i=1}^{m} a_{ij}\lambda_i = c_j, \qquad j \in I_X. \tag{1.10}$$

The second form of the optimality test for support programs can be stated as follows:

Optimality test. A support program X^* of problem (1.1)-(1.3) is optimal if

$$\sum_{i=1}^{m} a_{ij}\lambda_i \geqslant c_j \tag{1.11}$$

for $j=1, 2, \ldots, n$.

Proof. This sufficient condition for optimality of a support program is easily proved on the basis of the optimality criterion established in Chapter 3 (see Theorem 5.4). The above test also follows from the following equalities:

$$z_j = \sum_{i=1}^{m} c_{s_i} x_{ij} = \sum_{i=1}^{m} \Big(\sum_{\mu=1}^{m} a_{\mu s_i}\lambda_\mu\Big) x_{ij} = \sum_{\mu=1}^{m} \lambda_\mu \sum_{i=1}^{m} a_{\mu s_i} x_{ij} = \sum_{\mu=1}^{m} \lambda_\mu a_{\mu j}.$$

The first equality coincides with definition (1.5) of z_j. The second equality follows from (1.10). The third is obtained by reversing the order of summation. The fourth follows from (1.4) written for the components of the restraint vectors.

We now see that the relationship

$$\sum_{i=1}^{m} a_{ij}\lambda_i \geqslant c_j$$

for all j is equivalent to the condition

$$\Delta_j \geqslant 0 \text{ for } j=1, 2, \ldots, n.$$

We have thus proved not only the applicability of the second form of the optimality test, but also the equivalence of the two forms.

As we have already indicated, to each form of the optimality test corresponds a characteristic computational procedure used in solving linear-programming problems.

§ 2. General simplex procedure

2-1. We now consider the general procedure for solving the linear-programming problem in the nondegenerate case by means of the simplex method.

We remind the reader that a support program of problem (1.1)-(1.3) is said to be nondegenerate if the number of its positive components is m. A nondegenerate linear-programming problem is a linear-programming problem whose support programs are all nondegenerate.

Let a support program X of the problem whose basis is $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$, be given. The program is first tested for optimality. This can be done by either the first or the second form of the optimality test. In both cases we have to compute the Δ_j-values ($j=1, 2, \ldots, n$). In the first case Δ_j are

calculated from

$$\Delta_j = z_j - c_j = \sum_{i=1}^{m} c_{s_i} x_{ij} - c_j, \tag{2.1}$$

where x_{ij} are defined by (1.4). When the second form of the test is used, Δ_j are determined from

$$\Delta_j = z_j - c_j = \sum_{i=1}^{m} a_{ij} \lambda_i - c_j, \tag{2.2}$$

where $\Lambda = (\lambda_1, \ldots, \lambda_m)$ is the solution of system (1.10).

When testing a support program for optimality, one of the following three cases may arise:

a) $\Delta_j \geqslant 0$ for all $j \notin I_X$ (for $j \in I_X$ $z_j = c_j$, so that in the first case $\Delta_j \geqslant 0$ for all j from 1 to n).

b) $\Delta_j < 0$ for some single j, and all the $x_{ij} \leqslant 0$ $(i = 1, 2, \ldots, m)$ corresponding to this j are nonpositive.

c) $\Delta_j < 0$ for several j and for each of these j at least one of the x_{ij} is positive.

In the first case we have, from the optimality test of support programs, that X solves the linear-programming problem.

We shall now show that in case (b) the linear-programming problem is unsolvable, whereas in case (c) a method can be indicated for passing from the support program X to a new support program closer to the optimum.

2-2. Let us now see how the linear form (1.1) changes when we pass to a new program $X(\theta)$ formed from program X according to the following rules:

1. the j-th component of program $X(\theta)$ is taken equal to some positive number θ (j is a fixed index of an extrabasis variable of program X): $x_j(\theta) = \theta$;
2. the remaining extrabasis variables of program X are taken equal to zero ($x_s(\theta) = x_s = 0$ for $s \notin I_X$, $s \neq j$);
3. the basis components of $X(\theta)$ are chosen so that the vector $X(\theta) = (x_1(\theta), \ldots, x_n(\theta))$ defines a feasible program of the problem, i. e., the numbers $x_j(\theta)$ $(j = 1, 2, \ldots, n)$ satisfy restraints (1.2)–(1.3).

In the following we shall refer to this transition from program X to program $X(\theta)$ as the elementary transformation associated with the vector A_j.

Let us now write the relationships specifying the elementary transformation. From the definition of the support program X and from the assumptions made on the structure of program $X(\theta)$ we have

$$\sum_{i=1}^{m} x_{i0} A_{s_i} = B, \tag{2.3}$$

$$\sum_{i=1}^{m} x_{i0}(\theta) A_{s_i} + \theta A_j = B. \tag{2.4}$$

We recall that any restraint vector may be expressed in terms of basis vectors

$$A_j = \sum_{i=1}^{m} x_{ij} A_{s_i}. \tag{1.4}$$

From (1.4) and (2.3) we have

$$\sum_{i=1}^{m} (x_{i0} - \theta x_{ij}) A_{s_i} + \theta A_j = B. \tag{2.5}$$

Comparing (2.5) with (2.4) and since the solution of this system is unique

(the vectors $A_{s_1}, \ldots, A_{s_m}$ constituting the basis of the support program X are linearly independent), we obtain equalities defining the elementary transformation:

$$x_{i0}(\theta) = x_{i0} - \theta x_{ij}. \tag{2.6}$$

Here i specifies the position of the vectors A_{s_i} in the basis of the support program X $(x_{i0} = x_{s_i},\ x_{i0}(\theta) = x_{s_i}(\theta))$. We recall that according to the definition of the elementary transformation associated with vector A_j

$$x_t(\theta) = x_t = 0 \quad \text{for} \quad t \notin I_X, \quad t \neq j; \qquad x_j(\theta) = \theta.$$

The elementary transformation is thus well-defined.

All the x_{i0} — the basis components of program X— are positive. Hence, as long as θ is small enough for all the $x_{i0}(\theta)$ in (2.6) to be nonnegative, the vector $X(\theta)$ satisfies conditions (1.2), (1.3) and as such is a feasible program of the linear-programming problem.

Using the components of the feasible program $X(\theta)$ we easily calculate the corresponding value of the linear form (1.1):

$$L[X(\theta)] = \sum_{i=1}^{m} c_{s_i} x_{i0}(\theta) + c_j\theta = \sum_{i=1}^{m} c_{s_i} x_{i0} - \theta\Big(\sum_{i=1}^{m} c_{s_i} x_{ij} - c_j\Big).$$

Applying (1.5) and (1.6), we obtain

$$L[X(\theta)] = L(X) - \theta\Delta_j. \tag{2.7}$$

Equality (2.7) shows that the effect of elementary transformation of program X on the value of the linear form is determined by the sign of Δ_j. If $\Delta_j < 0$, the value of the linear form is increased; if $\Delta_j > 0$ it is decreased; if $\Delta_j = 0$ the linear form remains unchanged.

2-3. We now introduce several terms useful in the following.

The parameters Δ_j, whose signs determine the direction of change of the linear form under the elementary transformation associated with the vector A_j, should be called *evaluations of the restraint vectors* A_j *with respect to the given basis* (since the coefficients c_j of the linear form also enter the definition of Δ_j, it would be more exact to call the parameters Δ_j *evaluations of the augmented restraint vectors* $\bar{A}_j$ *with respect to the given basis*). The usefulness of this term is obvious from the following considerations. Elementary transformations associated with vectors corresponding to $\Delta_j < 0$ and $\Delta_j > 0$ respectively increase and decrease the value of the linear form. For $\theta = 1$ the change in the value of the linear form is exactly Δ_j. The parameters Δ_j thus evaluate the change which will take place in the value of the linear form when the vector A_j is introduced into the basis. It is this fact that is reflected in the proposed name of the parameters Δ_j. According to the preceding concept of basis-vector evaluation it would be better to define thus: $\Delta_j = c_j - z_j$ instead of $\Delta_j = z_j - c_j$. Nevertheless, to avoid all possible confusion, we shall keep to the latter definition which, by now, has become accepted in the literature on linear programming. So, $\Delta_j = z_j - c_j$. This will also enable us to unify the computational procedures.

When the second form of the optimality test is applied, Δ_j are calculated from

$$\Delta_j = \sum_{i=1}^{m} a_{ij}\lambda_i - c_j.$$

The parameters λ_i will, henceforth, be called *evaluations of problem restraints with respect to the given basis.*

To justify this term, let us recall the definition of the dual problem: the problem concerned with the minimization of the linear form $\sum_{i=1}^{m} b_i y_i$ subject to the conditions $\sum_{i=1}^{m} a_{ij} y_i \geqslant c_j$ is dual with respect to problem (1.1)–(1.3). It follows from the second form of the optimality test for support programs that the parameters λ_i^* associated with the basis of the support solution X^* satisfy the restraints of the dual problem

$$\Delta_j = \sum_{i=1}^{m} a_{ij} \lambda_i - c_j \geqslant 0.$$

On the other hand, for an optimal support program X^*

$$L(X^*) = \sum_{j=1}^{n} c_j x_j^* = \sum_{\mu=1}^{m} c_{s_\mu} x_{\mu 0}^* = \sum_{\mu=1}^{m} \left(\sum_{i=1}^{m} \lambda_i^* a_{is_\mu} \right) x_{\mu 0}^* = \sum_{i=1}^{m} \lambda_i^* \left(\sum_{\mu=1}^{m} a_{is_\mu} x_{\mu 0}^* \right) = \sum_{i=1}^{m} \lambda_i^* b_i.$$

Hence, the vector $\Lambda^* = (\lambda_1^*, \ldots, \lambda_m^*)$ solves the dual problem (see Chapter 3, Lemma 1.2).

In Chapter 3, 5-5 we interpreted the components of the optimal program of the dual problem as evaluations of the restraints of the primal problem (evaluations of the influence of the right-hand sides of (1.2) on the maximum obtainable value of the linear form (1.1)). By suitably choosing λ_i^*, all the Δ_j^* corresponding to the basis vectors of the solution vanish. The parameters λ_i corresponding to any support program play, with respect to the basis of this program, the same role as λ_i^* play, with respect to the basis of an optimal support program. We observe from the definition of λ_i that the choice of these parameters ensures that all the Δ_j— the evaluations of the restraint vectors of the basis — vanish.

The above considerations show that the parameters λ_i can indeed be interpreted as the evaluations of problem restraints with respect to the given basis.

2-4. With the aid of (2.6) and (2.7) it is possible to analyze all the cases which arise when testing a support program for optimality.

In case (a), as we have already seen, the program X solves the linear-programming problem according to the optimality test.

In case (b) there exists an index $j=k$ such that $\Delta_k < 0$ and all the corresponding components $x_{ik} \leqslant 0$ $(i = 1, 2, \ldots, m)$. From (2.6), for $j=k$, we see that in this case $X(\theta)$ is a program of problem (1.1)–(1.3) for any $\theta > 0$ $(x_j(\theta) \geqslant 0$ for any $\theta \geqslant 0)$. We observe from (2.7) that this indicates that the linear form $L(X)$ is unbounded above in the set of feasible programs of the problem. In case (b) the linear-programming problem is thus unsolvable.

Case (c), however, is the most frequent. In this case $\Delta_j < 0$ for several j, and for each of these j at least one of the x_{ij} is positive. We shall show that under these conditions, by means of the elementary transformation associated with the vector $A_k (\Delta_k < 0)$, it is possible to obtain from support program X another support program X' the value of whose linear form is higher.

We carry out the elementary transformation of program X with θ equal to the minimum of x_{i0}/x_{ik} for $x_{ik} > 0$, $i = 1, 2, \ldots, m$ (by assumption such x_{ik} exist).

Thus let

$$\theta_0 = \min_{\substack{i \\ x_{ik} > 0}} x_{i0}/x_{ik},$$

By assumption, $x_{i0}>0$ for $i=1, 2, \ldots, m$. Therefore θ_0 is positive. Here we use the fact that the support program X is nondegenerate.

Let $X'=X(\theta_0)$. By definition $X(\theta_0)$ is a feasible program of the problem. Moreover, we shall show that X' is a support program of the problem.

Let the minimum of x_{i0}/x_{ik}, defining θ_0, be attained for $i=r$. Obviously, $x'_{s_r}=x_{s_r}(\theta)=0$. We shall sometimes say that θ_0 is obtained on the r-th position of the basis, or, equivalently, on the restraint vector A_{s_r}.

According to (2.4), for given $\theta=\theta_0$ the constraint vector B is a linear combination of the vectors $A_{s_1}, A_{s_2}, \ldots, A_{s_{r-1}}, A_k, A_{s_{r+1}}, \ldots, A_{s_m}$ with nonnegative coefficients. It can easily be shown that this system of vectors is linearly independent. Indeed, we have the following theorem which will be used repeatedly in the following.

Theorem 2.1. If $P_1, P_2, \ldots, P_m$ is a linearly independent system of vectors and the vector $Q=\sum_{i=1}^{m}\zeta_iP_i$, where $\zeta_r\neq 0$, the system $P_1, P_2, \ldots, P_{r-1}, Q, P_{r+1}, \ldots$ $\ldots, P_m$ is also linearly independent.

Proof. Let us assume the contrary. Then, there exist numbers $\alpha_1, \alpha_2, \ldots$ $\ldots, \alpha_m, \alpha$, some of which are nonzero, such that

$$\sum_{\substack{i=1\\ i\neq r}}^{m}\alpha_iP_i+\alpha Q=0.$$

Expressing Q in terms of $P_1, P_2, \ldots, P_m$ and substituting we obtain

$$\sum_{\substack{i=1\\ i\neq r}}^{m}(\alpha_i+\alpha\zeta_i)P_i+\alpha\zeta_rP_r=0.$$

The system $P_1, \ldots, P_r, \ldots, P_m$ is, by assumption, linearly independent, and $\zeta_r\neq 0$. Therefore, the above relationship holds only if $\alpha=\alpha_1=\ldots=\alpha_{r-1}=$ $=\alpha_{r+1}=\ldots=\alpha_m=0$, which contradicts our assumption. This completes the proof.

From this theorem we have that $A_{s_1}, \ldots, A_{s_{r-1}}, A_k, A_{s_{r+1}}, \ldots, A_{s_m}$ is linearly independent (here the basis vectors constitute the system $P_1, \ldots, P_m$, $Q=A_k$, and $\zeta_r=x_{rk}>0$). This means that the feasible program $X'=X(\theta_0)$ is indeed a support program, its basis comprising the vectors $A_{s_1}, \ldots, A_{s_{r-1}}, A_k, A_{s_{r+1}}, \ldots, A_{s_m}$.

The basis of the new support program is, thus, obtained from the basis of the preceding support program when the vector A_{s_r} occupying the r-th position of the basis (the position on which θ_0 is obtained) is replaced by a vector A_k with negative relative evaluation. The vector A_k occupies the r-th position in the new basis. Accordingly,

$$x'_{i0}=\begin{cases} x_{i0}(\theta_0) & \text{for } i\neq r,\\ \theta_0 & \text{for } i=r.\end{cases}$$

Applying (2.6), we obtain expressions for the basis variables of the new support program:

$$x'_{i0}=\begin{cases} x_{i0}-\theta_0x_{ik}, & i\neq r,\\ \theta_0, & i=r.\end{cases} \tag{2.8}$$

Observe that in the nondegenerate problem θ_0 is obtained for a unique i, i. e., only for a single position of the basis. Otherwise the support program X' would be degenerate, since it would have had less than m positive components.

From (2.7)

$$L(X')=L(X)-\theta_0\Delta_k.$$

By assumption, $\Delta_k < 0$. Moreover, we have seen that θ_0 is positive. Therefore,

$$L(X') > L(X).$$

In case (c) we thus may transfer from the initial support program X to a new support program X' which is closer to the problem's solution. The transition to the new program increases the linear form by $-\theta_0\Delta_k$. Successive transitions from one support program to another are effected until either a solution is obtained, or unsolvability is established.

Each transformation from a given support program to another constitutes an iteration (step) of the simplex method. The number of iterations leading to a solution or to the proof of unsolvability is finite. Indeed, to each support program of the nondegenerate problem there corresponds a characteristic system of m linearly independent restraint vectors constituting its basis; there are altogether n restraint vectors. Therefore we have at most C_n^m different bases. The number of different support programs of the problem is thus also finite. For each support program the value of the linear form is uniquely determined. In the nondegenerate case each successive iteration increases the value of the linear form and, consequently, a basis which has once been used in an iteration will not be obtained a second time. Therefore, after a finite number of iterations the simplex method yields a solution (case (a)) or establishes unboundedness of the linear form in the set of feasible programs (case (b)).

Elementary transformation of support programs is fairly particular. Nevertheless, we have seen that elementary transformations invariably yield an optimal program of the problem in question. Solving a linear-programming problem by the simplex method involves determining suitable sequences of these transformations.

2-4. We review briefly the sequence of operations involved in each separate iteration of the simplex method.

Each iteration consists of two stages. In the first stage the given support program is tested for optimality. This stage leads to one of three possibilities (cases (a), (b), (c)). For case (a) or case (b), the process terminates. In case (a), the program in question is optimal. In case (b), unsolvability of the problem is established. For case (c), we proceed with the second stage of the iteration — determination of an elementary transformation which leads to a new support program with a higher value of the linear form. In the second stage a vector A_k with a negative evaluation ($\Delta_k<0$) is chosen and the position r to be occupied by this vector in the basis is determined. The vector A_{s_r} is eliminated from the basis. Thus, the new basis comprises the vectors of the old basis with the vector A_k replacing A_{s_r}. In the second stage all the parameters necessary for the optimality test of the new support program are computed. This new support program is then tested for optimality in the first stage of the next iteration.

All the arguments of the present section were stated for an linear-programming problem requiring maximization of the linear form (1.1) subject to conditions (1.2), (1.3). There is no need for a separate discussion in the case of minimization. Any problem involving minimization of the linear form

$$L(X) = \sum_{j=1}^{n} c_j x_j$$

subject to certain conditions is reducible to maximization of the

linear form

$$\tilde{L}(X) = -\sum_{j=1}^{n} c_j x_j$$

under the same conditions.

§ 3. Examples

3-1. We give below two simple examples illustrating the computational procedure for solving linear-programming problems by the simplex method using only formulas and results of the present chapter. More compact computational procedures are given in the next chapter. Here we consider only the principles of establishing an optimal program, not the technique of problem solution. The solution, therefore, is given with reference to the first form of the optimality test.

Example 1. Maximize the linear form

$$L(X) = 5x_1 - x_2 - 2x_3 + 5x_4 + 5x_5 - x_6$$

subject to the conditions

$$\left.\begin{array}{l} -2x_1 + 5x_2 + x_3 = 10 \\ x_1 - x_2 + x_4 = 1 \\ x_1 + 2x_2 + x_5 = 6 \\ 10x_1 - 3x_2 + x_6 = 15 \end{array}\right\}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, 6.$$

Solution. The restraint vectors and the constraint vector are, respectively,

$$A_1 = \begin{pmatrix} -2 \\ 1 \\ 1 \\ 10 \end{pmatrix}; \quad A_2 = \begin{pmatrix} 5 \\ -1 \\ 2 \\ -3 \end{pmatrix}; \quad A_3 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}; \quad A_4 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix};$$

$$A_5 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}; \quad A_6 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}; \quad B = \begin{pmatrix} 10 \\ 1 \\ 6 \\ 15 \end{pmatrix}.$$

It can be seen that as an initial support program we may take

$$X = (0, 0, 10, 1, 6, 15).$$

For this program $L(X) = 0$.

According to the general simplex procedure, we express A_1 and A_2 in terms of A_3, A_4, A_5 and A_6 which form the basis B of the initial support program. These vectors, respectively, occupy the positions 1, 2, 3, and 4 in the basis. For the sake of uniformity, we should, in accordance with the notations of § 1, take $A_3 = A_{s_1}$, $A_4 = A_{s_2}$, $A_5 = A_{s_3}$, $A_6 = A_{s_4}$. The expansion coefficients, x_{i1} and x_{i2} of the vectors A_1 and A_2 expressed in terms of the basis vectors satisfy the equations

$$A_1 = \sum_{i=1}^{4} x_{i1} A_{s_i},$$

$$A_2 = \sum_{i=1}^{4} x_{i2} A_{s_i}.$$

The basis vectors A_{s_i} are unit vectors. The components x_{i1} and x_{i2} are, therefore, equal to the corresponding components of the vectors A_1 and A_2. The coefficients x_{ij} for $j \in I_X$ are, obviously, equal to unity for $j = s_i$ and to zero for $j \neq s_i$. The matrix $\|x_{ij}\|$ thus coincides with the restraint matrix $\|a_{ij}\|$ (given different positions of the vectors in the basis, the sequence of rows in $\|x_{ij}\|$ and $\|a_{ij}\|$ would be different):

$$\|x_{ij}\| = \begin{Vmatrix} -2 & 5 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 & 1 & 0 \\ 10 & -3 & 0 & 0 & 0 & 1 \end{Vmatrix} = \|a_{ij}\|.$$

We now arrange the components of the support program X, which are the basis variables x_{i0}, according

to the positions of the vectors A_{s_i} in the basis:

$$x_{10}=10,\quad x_{20}=1,\quad x_{30}=6,\quad x_{40}=15.$$

The parameters z_j and Δ_j are computed from (1.5) and (1.6). We have

$$z=(z_1,\ \dots,\ z_6)=(4,\ -2,\ -2,\ 5,\ 5,\ -1),$$
$$\Delta=(\Delta_1,\ \dots,\ \Delta_6)=(-1,\ -1,\ 0,\ 0,\ 0,\ 0).$$

We see that Δ_1 and Δ_2 are negative and that to each correspond several positive x_{ij}. We thus have case (c), i. e., we may proceed, by means of the elementary transformation, from support program X to a new support program X' at the same time increasing the value of the linear form.

Either A_1 or A_2 may be introduced into the new basis B_1 ($\Delta_1=\Delta_2=-1$). Let us assume that A_1 is introduced into the basis. To establish which of the basis vectors should be eliminated, we calculate the ratios x_{i0}/x_{i1} for $x_{i1}>0$. We have $x_{11}<0$, whereas for $i=2,\ 3,\ 4$ the ratio is equal to 1; 6; $^{15}/_{2}$, respectively. The least ratio (= 1) corresponds to $i=2$. Hence, $\theta_0=1$ and the vector A_1 is introduced into position 2 of the basis (the position on which θ_0 is obtained) in place of the vector $A_{s_2}=A_4$. The basis components $x'_{i0}=x_{i0}(\theta_0)$ of the new program, re-enumerated according to the positions of the basis, are computed from (see (2.8))

$$x'_{i0}=\begin{cases} x_{i0}-1x_{i1} & \text{for } i\neq 2,\\ \theta_0=1 & \text{for } i=2.\end{cases}$$

The component $x'_4=x_4$ is again zero. The elementary transformation thus leads to a new support program:

$$X'=(1,\ 0,\ 12,\ 0,\ 5,\ 5).$$

On this program the linear form has increased from $L(X)=0$ to $L(X')=1$.

We now express A_2 and A_4 (vectors not appearing in the basis B_1) in terms of the vectors B_1. As before, we have to solve two systems of equations:

$$A_2=\sum_{i=1}^{4}x'_{i2}A_{s_i},$$
$$A_4=\sum_{i=1}^{4}x'_{i4}A_{s_i}.$$

In the new basis the vector A_1 is in position 2. The rest of the vectors of B_1 remain in the same positions as in the preceding basis:

$$A_{s_1}=A_3,\quad A_{s_2}=A_1,\quad A_{s_3}=A_5,\quad A_{s_4}=A_6.$$

Solving the two systems and since $x'_{ij}=\delta_j^{s_i}$ for $j\in I_{X'}$, we obtain (δ_j^i is the Kronecker delta equal to 1 when $j=i$ and to 0 when $j\neq i$)

$$\|x'_{ij}\|=\left\|\begin{matrix} 0 & 3 & 1 & 2 & 0 & 0\\ 1 & -1 & 0 & 1 & 0 & 0\\ 0 & 3 & 0 & -1 & 1 & 0\\ 0 & 7 & 0 & -10 & 0 & 1\end{matrix}\right\|.$$

As a check we may also determine the basis components of X' from

$$B=\sum_{j\in I_{X'}}x'_jA_j=\sum_{i=1}^{m}x'_{i0}A_{s_i}.$$

To the new support program X' correspond the vectors Z' and Δ'. Their components are computed in the same way as the components of the vectors Z and Δ:

$$Z'=(5,\ -3,\ -2,\ 6,\ 5\ -1),$$
$$\Delta'=(0,\ -2,\ 0,\ 1,\ 0,\ 0).$$

Here Δ'_2 is negative. Moreover, to Δ'_2 correspond several positive x'_{i2}. We thus may, by means of the elementary transformation, increase the value of the linear form. Subsequent iterations are carried out according to the same rules as the preceding ones. Omitting the intermediate computations, we arrange the components of the vectors $X^{(s)}$, $Z^{(s)}$ and $\Delta^{(s)}$ and the corresponding values of the linear form L in the following tableau (Table 4.1). We observe that all the components of the vector $\Delta^{(3)}$ are nonnegative. According to the optimality test, we have obtained an optimal program X^*– a solution of the linear-programming problem.

TABLE 4.1

s / i	2			3		
	$X^{(2)}$	$Z^{(2)}$	$\Delta^{(2)}$	$X^{(3)}$	$Z^{(3)}$	$\Delta^{(3)}$
1	12/7	5	0	48/23	5	0
2	5/7	—1	0	45/23	—1	0
3	69/7	—2	0	101/23	—2	0
4	0	22/7	—13/7	20/23	5	0
5	20/7	5	0	0	128/23	13/23
6	0	—5/7	2/7	0	—22/23	1/23
$L(X^{(s)})$	17/7			93/23		

3-2. We now give an example where the process terminates in case (b).

Example 2. Maximize the linear form

$$L(X)=3x_3+4x_4$$

subject to the conditions

$$\begin{aligned} 2x_1-3x_2+x_3 \quad &=1, \\ -5x_1+2x_2 \quad +x_4&=1, \\ x_i\geqslant 0, \quad i=1,\ \dots,\ 4.& \end{aligned}$$

Solution. It is easily seen that the vector

$$X=(0,\ 0,\ 1,\ 1)$$

may be taken as the initial support program. On program X, $L(X)=7$. The restraint vectors and the constraint vector of the problem are, respectively,

$$A_1=\begin{pmatrix}2\\-5\end{pmatrix};\quad A_2=\begin{pmatrix}-3\\2\end{pmatrix};\quad A_3=\begin{pmatrix}1\\0\end{pmatrix};\quad A_4=\begin{pmatrix}0\\1\end{pmatrix};\quad B=\begin{pmatrix}1\\1\end{pmatrix}.$$

The initial basis, B, comprises the unit vectors $A_3=A_{s_1}$ and $A_4=A_{s_2}$. Therefore

$$\|x_{ij}\|=\|a_{ij}\|=\left\|\begin{matrix}2 & -3 & 1 & 0\\ -5 & 2 & 0 & 1\end{matrix}\right\|.$$

Enumerating the basis components of the support program X according to the positions which the corresponding restraint vectors occupy in the basis we obtain

$$x_3=x_{10}=1,\quad x_4=x_{20}=1.$$

Applying (1.5) and (1.6), we compute the components of the vectors Z and Δ:

$$\begin{aligned} Z&=(-14,\ -1,\ 3,\ 4), \\ \Delta&=(-14,\ -1,\ 0,\ 0). \end{aligned}$$

To the negative evaluation $\Delta_1=-14$ corresponds a single positive x_{i1}, namely $x_{11}=2$. Therefore, A_1 is introduced in the basis in the first position of the basis, since the minimum ratio

$$\frac{x_{i0}}{x_{ik}}=\frac{x_{i0}}{x_{i1}}$$

is obtained for $i=1$. The vector A_3 occupying the first position of the basis is eliminated. The component x_1 of the new support program is

$$\theta_0=\frac{x_{10}}{x_{11}}=\frac{1}{2}.$$

The other components of X' are computed from (2.8). Here $x'_2 = x_2$ is again zero, since A_2, as before, does not appear in the basis.

We have thus obtained a new support program

$$X' = \left\{\frac{1}{2},\ 0,\ 0,\ \frac{7}{2}\right\}.$$

The value of the linear form corresponding to the support program X' is $L(X') = 14$.

The basis B_1 of the support program X' comprises the vectors A_1 and A_4. According to the positions of these vectors in the basis B_1, we have $A_1 = A_{s_1}$, $A_4 = A_{s_2}$ and $x'_1 = x'_{10}$, $x'_4 = x'_{20}$. We now express A_2 and A_3, which do not appear in B_1, in terms of the vectors of the new basis. The components x'_{i2} and x'_{i3} are determined from the equations

$$A_2 = \sum_{i=1}^{2} x'_{i2} A_{s_i},$$

$$A_3 = \sum_{i=1}^{2} x'_{i3} A_{s_i}.$$

Solving the two systems and since $x'_{ij} = \delta_j^{s_i}$ for $j \in I_{X'}$, we obtain

$$\| x'_{ij} \| = \begin{Vmatrix} 1 & -3/2 & 1/2 & 0 \\ 0 & -11/2 & 5/2 & 1 \end{Vmatrix}.$$

Hence

$$Z' = (0,\ -22,\ 10,\ 4),$$
$$\Delta' = (0,\ -22,\ 7,\ 0).$$

We see that $\Delta'_2 < 0$ and all the components $x'_{i2} < 0$. This relationship between the parameters Δ'_j and x'_{ij} corresponds to case (b), when the linear form has no upper bound in the set of feasible programs of the problem. The problem in question is thus unsolvable. Indeed, the point

$$X = (t,\ t,\ 1+t,\ 1+3t)$$

satisfies the problem restraints for any $t \geqslant 0$. Here

$$L(X) = 7 + 15t$$

increases to infinity as t increases.

§ 4. Geometrical interpretation of the simplex method

4-1. All the elements of the simplex method have a simple geometrical interpretation. In Chapter 2, § 5 we advanced two geometrical interpretations of the linear-programming problem. The reader is advised to reread the section dealing with the geometrical images associated with the algebraic description of the present chapter. The geometrical interpretation of the method will be given in terms of many-dimensional spaces. To understand the geometrical illustration well, the reader may confine the analysis to the two- or three-dimensional case. The examples in 4-2 and 4-4 are especillay designed for this purpose.

Different geometrical interpretations of the problem lead to different geometrical interpretations of the method.

According to the first geometrical interpretation of the linear-programming problem, restraints (1.2)–(1.3) define a convex polyhedron (or an unbounded convex polyhedral set) in the space of variables $x_1, \ldots, x_n$, the dimensionality of this set not exceeding $n-m$. This polyhedral set is contained in the intersection of the hyperplanes defined by restraints (1.2) of the problem.

To a support program X corresponds a vertex of the polyhedral set.

This vertex, like the support program, will be denoted by X. The vertex X is formed by the intersection of hyperplanes satisfying restraints (1.2) and hyperplanes corresponding to the nonzero components of the support program. In the nondegenerate case all the basis variables are nonzero. A support program contains $n-m$ extrabasis components. Correspondingly, in the nondegenerate case, the number of independent hyperplanes intersecting at the vertex X is

$$m+(n-m)=n.$$

Take one of the extrabasis variables x_j and replace the condition $x_j=0$ by the constraint $x_j\geqslant 0$. If all the other extrabasis variables retain their zero values, the locus obtained describes a ray "inside" the polyhedral set (where $x_j>0$). We shall say that this ray corresponds to the restraint vector A_j.

The locus of all points for which the linear form

$$L=\sum_{j=1}^{n} c_j x_j$$

maintains a constant value is a hyperplane. This will be called the linear-form hyperplane. The vector

$$C=(c_1, c_2, \ldots, c_n)$$

specifies the direction in which the hyperplane should be displaced to increase the value of L. The linear-form hyperplane generates two half-spaces in the $x_1, x_2, \ldots, x_n$-space. The half-space containing the vector C will be called the upper half-space; the other half-space is known as the lower half-space.

Geometrically, elementary transformation of the support program X associated with the vector A_j is a translation from the vertex X along the ray corresponding to the vector A_j. Any edge of the polyhedral set issuing from the vertex X is the intersection of some $n-1$ independent boundary hyperplanes containing the point X. Therefore, in the nondegenerate case, to any such edge there corresponds a ray determined by one of the vectors A_j, $j\notin I_X$. On the other hand, in our case the intersection of the ray corresponding to any vector A_j, $j\in I_X$, with the polyhedral set contains a segment $(\theta_0>0)$ and, consequently, constitutes an edge of the given set.

Thus, when the program X is nondegenerate, there is a one-to-one correspondence between the vectors A_j, $j\notin I_X$, and the edges of the polyhedral set issuing from the point X. We recall that edges of the polyhedral set may be either bounded or unbounded. The former are segments joining two vertices of the set, the latter are rays issuing from certain vertices.

We may now say that each set of elementary transformations with $0\leqslant\theta\leqslant\theta_0$ is geometrically equivalent to translation along an edge of the polyhedral set. If $\theta_0<\infty$ the corresponding edge is bounded. If $\theta_0=\infty$ the edge is not bounded.

For fixed θ, we move to the point $X(\theta)$, which modifies the linear form by $\Delta_j\theta$.

We now give in geometrical terms all the elements of a single iteration of the simplex method.

In the first stage of an iteration the support program X is tested for optimality. We draw through the vertex X the linear-form hyperplane and select all the edges of the polyhedral set issuing from the point X and

contained in the upper half-space. If there are no such edges, i. e., all the edges containing the vertex X are below the linear-form hyperplane, the support program X is optimal (case (a)). Geometrically it is obvious that if all the edges issuing from the vertex X of a polyhedral set are below the linear-form hyperplane, the entire set is below the hyperplane. This proves optimality.

Assume now that some edges of the polyhedral set lie in the upper half-space. Two possibilities arise. If among the edges in the upper half-space there is an unbounded edge (ray), the linear form obviously has no upper bound (case (b)). If, however, all the edges of the polyhedral set contained in the upper half-space are bounded (all the edges are segments), we may proceed to the second stage of the iteration, i. e., effect a translation along one of these edges thus increasing the value of the linear form (case (c)). Parallel translation of the linear-form hyperplane is continued until it intersects the other end point of the edge, the adjoining vertex X'. At X' one of the basis variables of the old basis vanishes. The vertex X'— the image of the new support program X'— is thus the intersection of the edge-forming hyperplanes with the hyperplane corresponding to the old basis variables which has now vanished. Obviously,

$$L(X') > L(X).$$

Proceeding with this parallel translation of the hyperplane $L(X) = \text{const}$ from a given vertex of the restraint polyhedron to another (adjoining) vertex, the value of the linear form is increased with each iteration. Since the number of vertices of the restraint polyhedron (polyhedral set) is finite, the maximum of the linear form is established after a finite number of steps. It is also obvious from geometrical considerations that unsolvability of the problem (unboundedness of the linear form in the domain of its definition), if the problem be indeed unsolvable, is also found after a finite number of steps.

4-2. We now illustrate the geometrical interpretation applying our reasoning to the examples given in § 3. We have already noted that the restraint polyhedron is contained in the intersection D of hyperplanes corresponding to the equality restraints of the problem. Introducing a system of coordinates in D we obtain the equivalent linear-programming problem with all restraints represented by inequalities. The equivalent problem can be obtained, for example, by expressing the basis variables of some support program in terms of the extrabasis variables and eliminating the basis variables from the restraints and from the linear form of the problem. Applying this procedure to the restraints of Example 1, we obtain

$$\begin{aligned} x_3 &= 10 + 2x_1 - 5x_2, \\ x_4 &= 1 - x_1 + x_2, \\ x_5 &= 6 - x_1 - 2x_2, \\ x_6 &= 15 - 10x_1 + 3x_2. \end{aligned}$$

We now state the equivalent problem.

Maximize the linear form

$$L(X) = x_1 + x_2$$

subject to the conditions

$$\begin{aligned} 10 + 2x_1 - 5x_2 &\geqslant 0, \\ 1 - x_1 + x_2 &\geqslant 0, \\ 6 - x_1 - 2x_2 &\geqslant 0, \\ 15 - 10x_1 + 3x_2 &\geqslant 0, \\ x_1 \geqslant 0, \quad x_2 &\geqslant 0. \end{aligned}$$

The domain of definition corresponding to the linear form of the equivalent problem (and consequently of Example 1) is shown in Figure 4.1. The sides of the restraint polygon are segments of the straight lines

$$x_1 = 0; \quad x_2 = 0;$$

$$x_3 = 10 + 2x_1 - 5x_2 = 0; \qquad x_5 = 6 - x_1 - 2x_2 = 0;$$

$$x_4 = 1 - x_1 + x_2 = 0; \qquad x_6 = 15 - 10x_1 + 3x_2 = 0.$$

Striation along the line indicates the half-plane in which the corresponding variable is positive. The lines $L(X)=\text{const}$ $(x_1+x_2=\text{const})$ are parallel to the line MN in Figure 4.1. The arrow OC indicates the direction in which the line MN should be translated to increase the value of the linear form. The initial support program corresponds to vertex O of the restraint polygon. The point O is the intersection of lines $x_1=0$ and $x_2=0$ corresponding to the extrabasis variables. The value of the linear form at this vertex is zero.

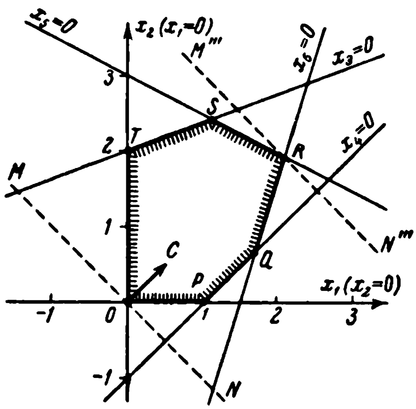


FIGURE 4.1

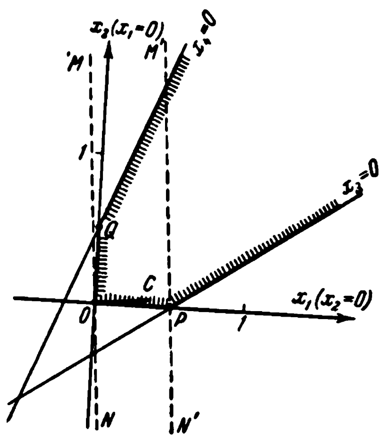


FIGURE 4.2

We observe from Figure 4.1 that all the sides (edges) of the polygon issuing from the vertex O are contained in the upper halfplane. Hence, translation to any of the adjoining vertices increases L. Elementary transformation of the support program X to program X' corresponds to translation from vertex O to vertex P of the restraint polygon. Translation is effected along the ray defined by the equation $x_2=0,\ x_1\geqslant 0$. The ray OP thus corresponds to the restraint vector A_1. At vertex P, the basis variable x_4 vanishes. Hence, the point P is the intersection of the lines $x_2=0,\ x_4=0$. The vertex P is the image of the support program X'.

Transition from program X' to program X'' and from program X'' to program X''' is realized by translation from vertex $P\,(x_2=0,\ x_4=0)$ to vertex $Q\,(x_4=0,\ x_6=0)$, and from this Q to vertex $R\,(x_6=0,\ x_5=0)$. Translation is effected along the rays corresponding to the restraint vectors $A_2(x_2\geqslant 0,\ x_4=0)$ and $A_4\,(x_4\geqslant 0,\ x_6=0)$, respectively. Transition to each successive vertex increases the value of the linear form.

All the sides (edges) of the polygon issuing from the vertex R are below the line $M'''N'''$. According to the geometrical interpretation of the optimality test, this means that the maximum of the linear form is obtained at vertex R. We observe from Figure 4.1 that any parallel translation of the line $M'''N'''$ ensuring nonempty intersection of this line with the restraint polygon decreases the linear form $L(X)$. The coordinates of the point R thus specify the optimal program of the problem.

In Figure 4.2 we show the domain of definition of the linear form of Example 2. We see that this domain is an unbounded convex polygonal figure. The lines $L(X)=\text{const}$ are parallel to the line MN. The vector C gives the direction in which the linear form increases. The initial support program X corresponds to vertex O of the polygonal figure. The value of the linear form $L(X)$ at the vertex O is 7. Translation to any adjoining vertex increases the value of L. The program X' corresponds to the vertex P. $L(X')=14$. The equation of the line $M'N'$ is

$$L(X')=14,$$

or

$$14x_1+x_2=7.$$

The ray $x_2\geqslant 0,\ x_3=0$ issuing from the vertex P is an unbounded edge of the polygonal set representing the domain of definition of the linear form. This ray is above the line $M'N'$. According to the geometrical interpretation of the simplex method, this establishes unsolvability of the problem. We see from Figure 4.2 that when $M'N'$ is translated parallel to the direction indicated by the vector C, it always intersects the domain of definition of the linear form. In the process $L(X)$ increases to infinity. We thus see that the linear form has no upper bound in the set of feasible programs.

4-3. We now give the geometrical interpretation of the simplex method corresponding to the second geometrical interpretation of the linear-programming problem.

We remind the reader that the augmented restraint vectors $\bar{A}_j$ generate a convex polyhedral cone K in the $(m+1)$-dimensional space of points $U=(u_1, u_2, \ldots, u_{m+1})$. To restraint vectors A_j and to the constraint vector B in the U-space correspond points with $u_{m+1}=0$. A line Q passes through the point defined by the constraint vector B, parallel to the Ou_{m+1}-axis. If the line Q does not intersect the cone K, the set of feasible programs is empty, i. e., the restraints are inconsistent.

We shall deal with cases where the set of feasible programs is nonempty, i. e., when the intersection of the line Q and the cone K is not empty. The part of the line Q contained in the cone K is the image of the domain of definition of the linear form in the $(m+1)$-dimensional space of points U. The highest point in the intersection of K and Q (the point of the segment with the greatest value of the $(m+1)$-th coordinate) specifies the maximum of the linear form L, and the lowest point gives the minimum of L (existence of these points is implied).

In some cases the intersection of the line Q and the cone K is a ray directed upwards or downwards. In this case the linear form is unbounded above or below on the set of feasible programs. If the entire line Q is contained in the cone K, the linear form is unbounded in either direction in the domain of its definition.

We shall now consider the geometrical constructions associated with each iteration of the simplex method.

Let the restraint vectors $A_1, A_2, \ldots, A_m$ constitute the basis of some support program X. The corresponding augmented vectors $\bar{A}_1, \ldots, \bar{A}_m$ are also linearly independent. Let Π denote the hyperplane generated by the vectors $\bar{A}_1, \ldots, \bar{A}_m$ and passing through the origin. This hyperplane is uniquely determined by the support program X and is indeed its image in the U-space. The hyperplane Π divides the augmented vectors $\bar{A}_{m+1}, \ldots, \bar{A}_n$ into two groups. One group is on the side of the hyperplane Π which contains the positive semiaxis Ou_{m+1}. We shall say that these vectors are above the hyperplane Π. The second group contains all the vectors $\bar{A}_j$ which are below Π.

As before, we shall use the same symbol to denote vectors and the corresponding points. Let the lines passing through the points A_j and $\bar{A}_j$ (parallel to Ou_{m+1}) intersect the hyperplane Π at the points A_j^0, and the line Q, at the point B^0.

We now express the lengths of the segments $A_j^0 A_j$, $\bar{A}_j A_j^0$ and $B^0 B$ in terms of the parameters of the simplex method. This will enable us to relate the geometrical constructions with the algebraic description of the method given above.

By definition, the point A_j^0 belongs to the hyperplane Π. We express the vector A_j^0 in terms of the m linearly independent augmented restraint vectors $\bar{A}_i$ spanning the hyperplane Π:

$$A_j^0 = \sum_{i=1}^{m} \tilde{x}_{ij} \bar{A}_i. \tag{4.1}$$

The first m components of the vectors A_j^0 and $\bar{A}_i$ specify the restraint vectors A_j and A_i, respectively. We have that (see (1.4))

$$A_j = \sum_{i=1}^{m} x_{ij} A_i.$$

Therefore

$$\tilde{x}_{ij} = x_{ij}. \tag{4.2}$$

The length of the directional segment $A_j^{\circ}A_j$ is actually the $(m+1)$-th component of the vector A_j°*.

The last component of the augmented restraint vector $\bar{A}_i$ is c_i. We therefore have from (4.1), (4.2), and (1.5)

$$|A_j^{\circ}A_j| = \sum_{i=1}^{m} x_{ij}c_i = z_j \tag{4.3}$$

and

$$|\bar{A}_jA_j^{\circ}| = |\bar{A}_jA_j| - |A_j^{\circ}A_j| = c_j - z_j = -\Delta_j. \tag{4.4}$$

The segment $B^{\circ}B$ determines the value of the linear form for the given support program:

$$|B^{\circ}B| = L(X). \tag{4.5}$$

Thus, for the augmented vectors of the first group, those above the hyperplane Π, $\Delta_j < 0$. For the vectors $\bar{A}_j$ of the second group, $\Delta_j > 0$.

Let $\bar{A}_j$ be an augmented restraint vector not contained in the hyperplane Π. Consider the $(m+1)$-dimensional cone spanned by the vectors $\bar{A}_1, \bar{A}_2, \ldots$ $\ldots, \bar{A}_m, \bar{A}_j$. We shall denote this cone by K_j (j being the index of the vector $\bar{A}_j \notin \Pi$). The intersection of the line Q and the cone K_j comprises points which are geometrical images of programs which are obtained from X by the elementary transformation associated with the vector $A_j (0 \leqslant \theta \leqslant \theta_0)$. In the nondegenerate case the point X is situated inside the face spanned by the vectors $\bar{A}_1, \ldots, \bar{A}_m$. The intersection of Q and K_j for any $\bar{A}_j \notin \Pi$ therefore contains a certain segment. In the degenerate case the intersection of Q and K_j may contain only one point (Q tangent to K_j).

In the first stage of the iteration the support program is tested for optimality. The geometrical test reduces to deciding whether there exist augmented vectors above the hyperplane Π. If the first group of augmented vectors is an empty set, the support program X is optimal (case (a)). In this case the entire cone K is below the hyperplane Π. The maximum value of the linear form is determined by the length of the segment $B^{\circ}B$.

Let the first group of vectors $\bar{A}_j$ be nonempty, i. e., there exist augmented restraint vectors above the hyperplane Π. Here two possibiliites may arise corresponding to cases (b) and (c), respectively.

Geometrically, case (b) obtains when there exists a vector A_j above the hyperplane Π such that the intersection of K_j and Q is a ray issuing from the point X. We observe that this is possible if and only if the cone K_j contains the halfaxis Ou_{m+1} (see Exercise 3). If the first group of augmented restraint vectors is nonempty, and none of the K_j contains Ou_{m+1}, we have case (c). In this case we may pass from Π to Π', corresponding to a successive support program X'.

The hyperplane Π' intersects the line Q at the point B' above the point B°.

The transformation from Π to Π' is effected as follows. Let the vector A_k be above the hyperplane Π. We form the cone K_k. As we have already observed, in the nondegenerate case the intersection of K_k and Q contains a certain segment. In case (c) the intersection of K_k and Q may not be a ray. It therefore comprises a segment whose end points are the highest and the lowest points of intersection of K_k and Q. The point B° is the

* Here the "length" $\|AB\|$ of a directional segment AB is taken as the length of the segment AB with the corresponding sign. $|AB|>0$ ($|AB|<0$) if A is above (below) B in the sense of the Ou_{m+1}-axis.

lowest point of the segment. Let B' denote the highest point of the segment. Observe that the $(m+1)$-dimensional cone K_k has $(m+1)$ m-dimensional faces. Each of these faces is a cone spanned by some m vectors of the system $\overline{A}_1, \ldots, \overline{A}_m, \overline{A}_k$. Let the face of the cone K_k not containing the vector A_j $(j=1, 2, \ldots, m, k)$ be denoted by Γ_{jk}. In the nondegenerate case, the point B' lies inside some face of the cone K_k, say Γ_{rk}.

Let Π' be the hyperplane containing the face Γ_{rk}. The hyperplane Π' is the geometrical image of the new support program X'. Obviously, the basis of program X' comprises the vectors $\overline{A}_1, \overline{A}_2, \ldots, \overline{A}_{r-1}, \overline{A}_{r+1}, \ldots. \overline{A}_m, \overline{A}_k$, which enter the system spanning the face Γ_{rk}. The line Q intersects the hyperplane Π' at point B'. The point B' is above the point B^0, so that the program X' corresponds to a higher value of the linear form of the problem than problem X. This completes the iteration. The successive iteration is effected according to the same rules. We again check whether the hyperplane Π' is above the cone K. If the answer is in the positive, the hyperplane Π' specifies an optimal program, and the point B' determines the maximum value of the linear form. Otherwise, we should pass from hyperplane Π' to hyperplane Π'' following the same rules as in the transformation from Π to Π'. In the nondegenerate case we invariably obtain a hyperplane Π^* above the cone K, or establish unsolvability of the problem.

4-3. Let us now illustrate the solution of the following problem by geometrical constructions.

Maximize the linear form

$$L = 10x_1 + 8x_2 + 7x_3 + 16x_4 + 21x_5$$

subject to the conditions

$$\begin{gathered} 4x_1 + 2x_2 + 5x_3 + 10x_4 + 5x_5 = 6, \\ 9x_1 + 10x_2 + 12.5x_3 + 18x_4 + 16.5x_5 = 14, \\ x_j \geqslant 0, \ j = 1, 2, \ldots, 5. \end{gathered}$$

Figure 4.3 shows restraint vectors A_j, the constraint vector B, the corresponding augmented vectors, and the line Q passing through the point B parallel to the Ou_3-axis. We see that the line Q intersects the convex polyhedral cone spanned by the augmented restraint vectors. Hence, the set of feasible programs is nonempty.

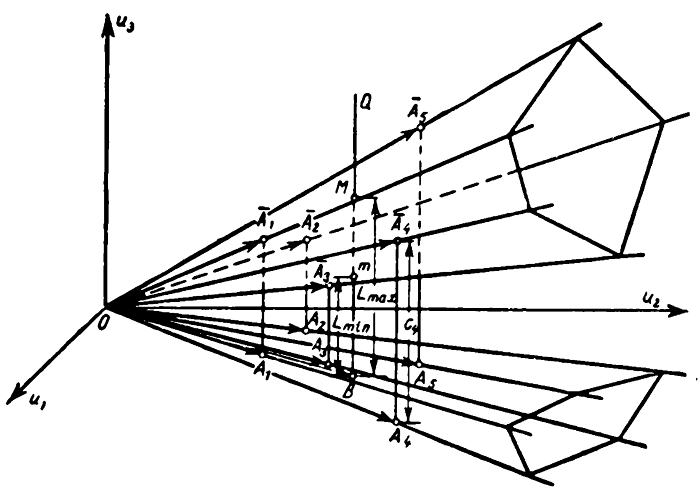


FIGURE 4.3

The vectors A_3 and A_4 are linearly independent, and the vector B lies inside the cone (here, plane angle) spanned by these vectors. This indicates that the constraint vector can be written as a linear combination of the vectors A_3 and A_4 with positive coefficients. Thus, it is certainly justifiable to take the restraint vectors

A_3 and A_4 as the basis of the initial support program X. The two-dimensional cone spanned by the vectors A_3 and A_4 is striated in Figure 4.4. The hyperplane Π (in our example, ordinary plane) is defined by the two-dimensional cone spanned by the augmented restraint vectors $\bar{A}_3$ and $\bar{A}_4$. In Figure 4.4 this cone is also striated.

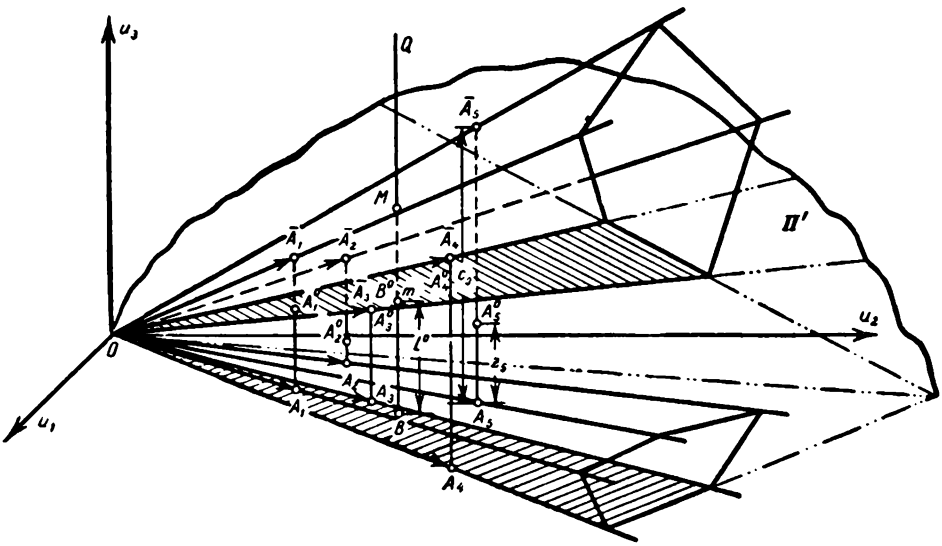


FIGURE 4.4

By construction, the points $\bar{A}_j$ are at a distance c_j from the u_1Ou_2-plane. In our problem

$$C=\{c_j\}=(10,\ 8,\ 7,\ 16,\ 21).$$

Perpendiculars dropped from the points $\bar{A}_j$ to the u_1Ou_2-plane intersect Π at points A_j^0, respectively. As we have seen, the "lengths" of the segments $A_j^0\, A_j$ are, respectively, equal to $\bar{z}_j$, and

$$|B^0B|=L\,(X).$$

In our case

$$Z=\{z_j\}=\left(\frac{206}{35},\ \frac{48}{35},\ 7,\ 16,\ \frac{41}{7}\right),$$

$$L\,(X)=\frac{304}{35}.$$

The "length" of the segment $\bar{A}_j\, A_j^0$ is $c_j-z_j=-\Delta_j$. In our example

$$\Delta=\left\{\Delta_j\right\}=\left(-\frac{144}{35},\ -\frac{232}{35},\ 0,\ 0,\ -\frac{106}{7}\right),$$

where Δ_1, Δ_2 and Δ_5 are negative.

Consequently, the points $\bar{A}_1$, $\bar{A}_2$ and $\bar{A}_5$ are above the plane Π. The point $\bar{A}_5$ is the highest of the set ($\Delta_5<\Delta_j,\ j=1,\ 2,\ 3,\ 4$). It is, therefore, expedient to introduce the restraint vector A_5 into the basis.

We observe from Figure 4.4 that the constraint vector B lies inside the cone (plane angle) spanned by the vectors A_4 and A_5, but outside the cone spanned by the vectors A_3 and A_5. This indicates that the coefficients in the expression of the constraint vector B in terms of A_4 and A_5 are positive, whereas those in the expression of B in terms of A_3 and A_5 may also be negative. We thus conclude that A_3 should be eliminated from the basis.

We arrive at the same conclusion also from the following considerations, which have been previously given for the general case. The line Q intersects the three-dimensional cone K_5 spanned by the vectors $\bar{A}_3$ and $\bar{A}_4$ (corresponding to the basis of the support program X) and the vector A_5 (corresponding to the restraint vector introduced into the basis) at two points: B^0 of the two-dimensional face Γ_{55} defined by the vectors $\bar{A}_3$ and $\bar{A}_4$ and B' of the face Γ_{35} and defined by the vectors $\bar{A}_4$ and $\bar{A}_5$. The point B^0 belongs to the plane Π generated by the vectors of the initial basis. The point B' is inside the plane Π' spanned by the vectors of the new basis. Again A_3 should be eliminated.

Similar constructions give the points A_j' and B'. Here

$$-\Delta' = C - Z' = \{c_j - z_j'\} = \{|\bar{A}_j A_j| - |A'_j A_j|\} =$$
$$= \left\{ \left| \bar{A}_j A_j' \right| \right\} = \left(\frac{12}{75}, \ -\frac{472}{75}, \ -\frac{106}{15}, \ 0, 0 \right);$$
$$L(X') = \frac{1136}{75}.$$

We observe from Figure 4.5 that at this step only one point $\bar{A}_j'$, namely $\bar{A}_1$, is above the plane Π' ($\Delta_1 < 0$). Hence, to increase the value of the linear form the vector A_1 must be introduced into the basis. Reasoning as

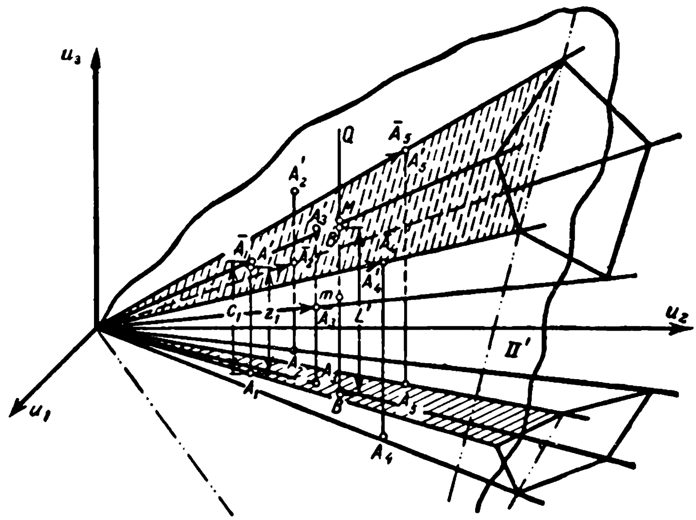


FIGURE 4.5

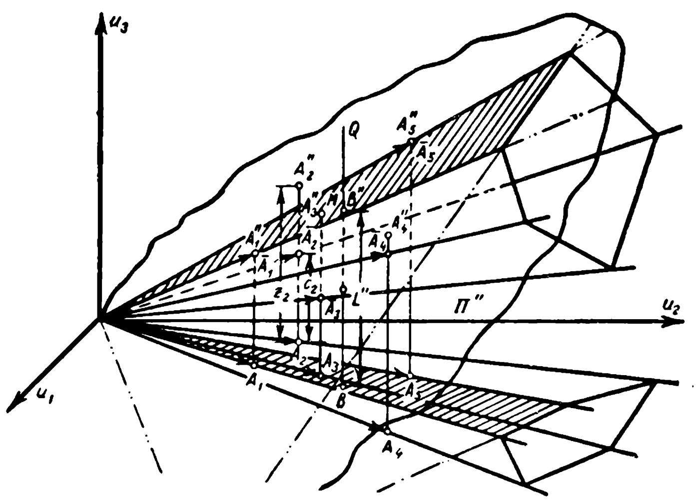


FIGURE 4.6

above we conclude that the vector A_4 should be eliminated from the basis (the constraint vector B lies outside the angle spanned by the vectors A_1 and A_4). The vectors $\bar{A}_1$ and $\bar{A}_5$ define the plane Π'' (Figure 4.6) above the cone K. The support program X'' corresponding to the basis (A_1, A_5) is thus a solution of the problem.

Expressing the constraint vector B in terms of the basis vectors, we obtain the optimal program

$$X'' = \left(\frac{29}{21},\ 0,\ 0,\ 0,\ \frac{2}{21}\right).$$

The line Q intersects the plane Π'' at the point

$$u_3 = L(X'') = \frac{332}{21}.$$

This is the maximum value of the linear form.

§ 5. The case of bilateral restraints

5-1. Consider the following linear-programming problem.
Maximize the linear form

$$L(X) = \sum_{j=1}^{n} c_j x_j \tag{5.1}$$

subject to the conditions

$$\sum_{j=1}^{n} A_j x_j = B, \tag{5.2}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j; \qquad j = 1, 2, \ldots, n. \tag{5.3}$$

The matrix $\|a_{ij}\|_{mn} = (A_1, A_2, \ldots, A_n)$ is assumed to be of rank m.

Problem (5.1)-(5.3) differs from the general linear-programming problem (1.1)-(1.3) written in canonical form only in restraints (5.3) which, generally speaking, impose upper and lower bounds on the variables. Problems with bilateral restraints are often encountered in practical applications and they therefore deserve special discussion.

We shall not assume that all the variables of the problem are bilaterally constrained by (5.3); some of them may have only one bound. If the variable x_j is bounded only above, $\alpha_j = -\infty$; if x_j is bounded only below, $\beta_j = \infty$. In particular, if $\alpha_j = 0$ and $\beta_j = \infty$, problem (5.1)-(5.3) reduces to the general linear-programming problem in canonical form.

Problem (5.1)-(5.3) can be reduced to canonical form. To this end it suffices to replace the variables x_j by $x_j' = x_j - \alpha_j$, $j = 1, 2, \ldots, n$, imposing nonnegativity requirements on the variables x_j' for $j = n+1, \ldots, 2n$. (Here we assume all α_j and β_j to be finite.)

Thus, a problem with bilateral restraints can be solved in two stages. Problem (5.1)-(5.3) is first reduced to canonical form and then solved by the method described in the preceding sections for problem (1.1)-(1.3). This method, however, is not practicable. The canonical form of the problem with bilateral restraints has $2n$ variables and $n+m$ equality restraints. In solving this problem we therefore have to deal with $(n+m)$-dimensional restraint vectors. Each iteration becomes most cumbersome.

If each variable of problem (5.1)-(5.3) is unilaterally bounded (above or below), the problem is easily reduced to canonical form by the following substitution of variables

$$x_j' = \begin{cases} x_j - \alpha_j, & \text{if } x_j \geqslant \alpha_j, \\ \beta_j - x_j, & \text{if } x_j \leqslant \beta_j. \end{cases}$$

The additional restraints distinguishing the problem in question from

problem (1.1)–(1.3) are, thus, very simple in structure:

$$x_j \leqslant \beta_j,$$

where x_j is any bilaterally restrained variable of problem (5.1)–(5.3). The simple structure of the additional restraints indicates that the simplex method can probably be extended to problems with bilateral restraints without substantial difficulties. In the following we shall show that this is indeed so.

Notice that by taking into consideration the specific features of the problem and adapting the linear-programming methods to the natural form of the problem, the computations associated with the determination of optimal programs are as a rule greatly simplified. In the following we shall often modify the computational methods of linear programming when applying them to various particular classes of problems. This modification is generally best achieved by applying the method taking into account the general procedure and the specific features of the problem. We shall do precisely this.

5-2. We first introduce some definitions. The vectors $A_j=(a_{1j}, \ldots, a_{mj})^T$, $j=1, \ldots, n$ are called restraint vectors of problem (5.1)–(5.3), and the vector $B=(b_1, \ldots, b_m)^T$ is the constraint vector of this problem.

A feasible program $X=(x_1, x_2, \ldots, x_n)$ of problem (5.1)–(5.3) is called a support program if the system of restraint vectors A_j corresponding to the components x_j for which

$$\alpha_j < x_j < \beta_j, \tag{5.4}$$

is linearly independent.

Observe that this definition of a support program is fully consistent with the concept of a support program introduced in Chapter 2, 4-2 for an arbitrary linear-programming problem. The proof of this is left to the reader (see Exercise 4).

Let X be a support program of problem (5.1)–(5.3). Consider the set of all the restraint vectors to which the components x_j of program X satisfying inequality (5. 4) correspond. The system of m linearly independent vectors containing this set is called the basis of the support program X. The components of the support program corresponding to the basis vectors will be called basis variables, the remaining components being extrabasis variables. An extrabasis variable x_j is, obviously, equal to one of the limiting values of the corresponding program component (α_j or β_j).

5-3. We now formulate the optimality test which is the basis of solving problems with bilateral restraints.

Let $X=(x_1, \ldots, x_n)$ be a support program of problem (5.1)–(5.3), its basis comprising the vectors $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$. As previously, let I_X denote the set of indices of the basis vectors. Applying the notations of § 1, we have

$$A_j = \sum_{i=1}^{m} x_{ij} A_{s_i}, \qquad j=1, 2, \ldots, n; \tag{5.5}$$

$$z_j = \sum_{i=1}^{m} x_{ij} c_{s_i}, \qquad j=1, 2, \ldots, n. \tag{5.6}$$

Let, as in § 2,

$$x_{i0} = x_{s_i}, \qquad i=1, 2, \ldots, m,$$

and let us introduce a vector A_0 satisfying the equality

$$A_0 = B - \sum_{j \notin I_X} x_j A_j = \sum_{i=1}^{m} x_{i0} A_{s_i}.$$

Combining the above relationships with (5.5), we obtain

$$A_j = \sum_{i=1}^{m} x_{ij} A_{s_i}, \qquad j = 0, 1, 2, \ldots, n. \tag{5.5'}$$

We remind the reader that the first subscript in the coefficients x_{ij} (the index i) indicates the position of the vector A_{s_i} in the basis.

Now let the parameters Δ_j be given by

$$\Delta_j = \begin{cases} z_j - c_j, & \text{if } \alpha_j \leqslant x_j < \beta_j, \\ c_j - z_j, & \text{if } x_j = \beta_j. \end{cases} \tag{5.7}$$

Obviously, for $j \in I_X$, in particular, with $\alpha_j < x_i < \beta_j$,

$$\Delta_j = 0.$$

In the following we shall see that Δ_j are analogs of the relative evaluations of the restraint vectors of problem (1.1)–(1.3).

Definition (5.7) of the parameters Δ_j makes it possible to state the optimality test for support programs of problem (5.1)–(5.3) in the same form as for problem (1.1)–(1.3).

Optimality test. A support program X is a solution of problem (5.1)–(5.3) if $\Delta_j \geqslant 0$ for all $j \notin I_X$.

Proof. Let $X' = (x_1', \ldots, x_n')$ be a support program of problem (5.1)–(5.3). We have

$$L(X') = \sum_{j=1}^{n} c_j x_j' = \sum_{j \in I_X} c_j x_j' + \sum_{\alpha} c_j x_j' + \sum_{\beta} c_j x_j',$$

where $\sum_{\alpha} (\sum_{\beta})$ are summations over all $j \notin I_X$ for which $x_j = \alpha_j$ ($x_j = \beta_j$).

Applying (5.3) and the conditions of the optimality test, we obtain

$$\sum_{\alpha} c_j (x_j' - \alpha_j) \leqslant \sum_{\alpha} z_j (x_j' - \alpha_j),$$
$$\sum_{\beta} c_j (x_j' - \beta_j) \leqslant \sum_{\beta} z_j (x_j' - \beta_j).$$

Therefore

$$\sum_{j \notin I_X} c_j (x_j' - x_j) \leqslant \sum_{j \notin I_X} z_j (x_j' - x_j).$$

This inequality gives the following estimate for the value of the linear form at point X':

$$L(X') \leqslant \sum_{j \in I_X} c_j x_j' + \sum_{j \notin I_X} z_j (x_j' - x_j) + \sum_{j \notin I_X} c_j x_j.$$

Further, applying (5.6) we have

$$L(X') \leqslant \sum_{i=1}^{m} c_{s_i} \Big[x_{s_i}' + \sum_{j \notin I_X} x_{ij} (x_j' - x_j)\Big] + \sum_{j \notin I_X} c_j x_j. \tag{5.8}$$

We shall now show that

$$x_{s_i}' + \sum_{j \notin I_X} x_{ij} (x_j' - x_j) = x_{s_i} = x_{i0}, \qquad i = 1, 2, \ldots, m. \tag{5.9}$$

Indeed, X' is a feasible program. Therefore

$$\sum_{i=1}^{m} A_{s_i} x_{s_i}' = B - \sum_{j \notin I_X} A_j x_j'. \tag{5.10}$$

Multiplying both sides of (5.5) by $x'_j - x_j$ and adding the results obtained for $j \in I_X$ to (5.10), we obtain

$$\sum_{i=1}^{m} A_{s_i}\left[x'_{s_i} + \sum_{j \notin I_X} x_{ij}(x'_j - x_j)\right] = B - \sum_{j \notin I_X} A_j x_j = A_0.$$

Let us compare this equality with (5.5') for $j=0$. The vectors A_{s_i} ($i=1, 2, \ldots$ $\ldots, m$) constitute the basis of the support program X. Linear independence of this system of vectors establishes the validity of (5.9).

From (5.8) and (5.9) we have

$$L(X') \leqslant \sum_{i=1}^{m} c_{s_i} x_{i0} + \sum_{j \notin I_X} c_j x_j = \sum_{j=1}^{n} c_j x_j = L(X). \tag{5.11}$$

Formula (5.11) applies for any X'. This indicates that X is an optimal program of problem (5.1)–(5.3). This proves the optimality test.

In § 1 we established the following relationship:

$$z_j = \sum_{i=1}^{m} \lambda_i a_{ij}, \qquad j = 1, 2, \ldots, n, \tag{5.12}$$

where the vector $\Lambda = (\lambda_1, \ldots, \lambda_m)$ is determined from the system of equations

$$\sum_{i=1}^{m} a_{ij}\lambda_i = c_j \quad \text{for} \quad j \in I_X. \tag{5.13}$$

Thus, the parameters z_j can be computed either from (5.6) or from (5.12). Depending on the formula chosen for computing z_j, required in determining Δ_j, we shall distinguish, as in § 1, between two forms of the optimality test. In the first form of the test, z_j are computed from (5.6), and in the second form from (5.12). The second form of the optimality test can be stated as follows.

Optimality test. A support program $X = (x_1, \ldots, x_n)$ is optimal if the vector Λ determined from (5.13) satisfies the inequalities

$$(\Lambda, A_j) = \sum_{i=1}^{m} a_{ij}\lambda_i \geqslant c_j, \qquad \textit{if} \quad x_j = \alpha_j,$$
$$(\Lambda, A_j) = \sum_{i=1}^{m} a_{ij}\lambda_i \leqslant c_j, \qquad \textit{if} \quad x_j = \beta_j.$$

5-4. To describe the general procedure for solving problem (5.1)–(5.3) it is expedient to extend the concept of elementary transformation introduced in § 2 to the case of bilaterally restrained variables. In a problem with bilateral restraints the elementary transformation associated with the vector A_j ($j \notin I_X$) is defined differently depending on the actual limit (α_j or β_j) with which the extrabasis variable x_j coincides.

Let us consider the two cases separately:

(1) $x_j = \alpha_j$, $j \notin I_X$,

(2) $x_j = \beta_j$, $j \notin I_X$.

1. In the first case, the vector A_j is associated with several transformations of the support program X to a program $X(\theta) = (x_1(\theta), \ldots, x_n(\theta))$ defined as

$$x_\mu(\theta) = \begin{cases} x_{s_i} - \theta x_{ij}, & \text{if} \quad \mu = s_i,\ i = 1, 2, \ldots, m, \\ x_j + \theta, & \text{if} \quad \mu = j, \\ x_\mu, & \text{if} \quad \mu \notin I_X,\ \mu \neq j. \end{cases} \tag{5.14}$$

We write the conditions under which the vector $X(\theta)$ is a program of problem (5.1)–(5.3).

Applying (5.5), we have

$$\sum_{j=1}^{n} x_j(\theta) A_j = \sum_{j=1}^{n} x_j A_j + \theta \left[A_j - \sum_{i=1}^{m} x_{ij} A_{s_i}\right] = \sum_{j=1}^{n} x_j A_j = B$$

for any θ. Hence, the vector $X(\theta)$ satisfies restraints (5.2) for any θ.

We now determine the set of all θ for which all the components of the vector $X(\theta)$ satisfy (5.3).

Since

$$x_j(\theta) = x_j + \theta = \alpha_j + \theta,$$

the inequalities

$$\alpha_j \leqslant x_j(\theta) \leqslant \beta_j$$

are satisfied if and only if

$$0 \leqslant \theta \leqslant \beta_j - \alpha_j = \beta_j - x_j. \tag{5.15}$$

The other restraints imposed on θ are determined by

$$\alpha_{s_i} \leqslant x_{s_i}(\theta) \leqslant \beta_{s_i} \text{ for } i = 1, 2, \dots, m. \tag{5.16}$$

Obviously, these inequalities need not hold only if $x_{ij} \neq 0$.

If $x_{ij} > 0$,

$$x_{s_i}(\theta) = x_{s_i} - \theta x_{ij} \leqslant x_{s_i} \qquad (\theta \geqslant 0).$$

Hence, for $x_{ij} > 0$, the i-th inequality of (5.16) is satisfied if and only if

$$\theta \leqslant \frac{x_{s_i} - \alpha_{s_i}}{x_{ij}}. \tag{5.17}$$

If $x_{ij} < 0$, then for $\theta > 0$

$$x_{s_i}(\theta) = x_{s_i} - \theta x_{ij} \geqslant x_{s_i}.$$

Hence, for $x_{ij} < 0$, the i-th inequality of (5.16) is satisfied if

$$\theta \leqslant \frac{\beta_{s_i} - x_{s_i}}{-x_{ij}} = \frac{x_{s_i} - \beta_{s_i}}{x_{ij}}. \tag{5.18}$$

Combining (5.15), (5.17), and (5.18), we see that $X(\theta)$ satisfies all the restraints (5.13) if and only if

$$0 \leqslant \theta \leqslant \theta_0, \tag{5.19}$$

where θ_0 is the least of the three numbers

$$\theta_0^{(1)} = \min_{x_{ij} > 0} \frac{x_{s_i} - \alpha_{s_i}}{x_{ij}},$$

$$\theta_0^{(2)} = \min_{x_{ij} < 0} \frac{x_{s_i} - \beta_{s_i}}{x_{ij}},$$

$$\theta_0^{(3)} = \beta_j - x_j = \beta_j - \alpha_j.$$

Until now we have considered m positions of the basis, each position representing one of the basis vectors. To simplify notations, we find it convenient to introduce the $(m+1)$-th position in which the vector $A_j = A_{s_{m+1}}$, which specifies the given elementary transformation, stands. This additional position will be of use in the following also.

Let

$$x_{m+1,j} = -1; \quad x_{m+1,0} = x_j = x_{s_{m+1}}.$$

If we take

$$\gamma_i = \begin{cases} \alpha_{s_i} \text{ for } x_{ij} > 0, & i = 1, 2, \dots, m+1, \\ \beta_{s_i} \text{ for } x_{ij} < 0, & i = 1, 2, \dots, m+1, \end{cases} \tag{5.20}$$

the formula for θ_0 can be written in the following compact form:

$$\theta_0 = \min_{\substack{x_{ij} \neq 0 \\ 1 \leqslant i \leqslant m+1}} \frac{x_{i0} - \gamma_i}{x_{ij}}. \tag{5.21}$$

Here, as before, $x_{i0} = x_{s_i}$ is the basis variable occupying the i-th position in the basis.

The elementary transformation associated with the vector A_j for which $x_j = \alpha_j$ ($j \notin I_X$) is, thus, defined as transformation (5.14) with θ satisfying (5.19). The parameter θ_0 in inequality (5.19) is determined from (5.21).

2. We now introduce the concept of elementary transformation for the second case.

In this case, to vector A_j, associated with the elementary transformation, there corresponds an extrabasis variable $x_j = \beta_j$.

The set of elementary transformations of program X to program $X(\theta)$ associated with the vector A_j is given by

$$x_\mu(\theta) = \begin{cases} x_{s_i} + \theta x_{ij}, & \text{if } \mu = s_i, \quad i = 1, 2, \ldots, m, \\ x_j - \theta, & \text{if } \mu = j, \\ x_\mu, & \text{if } \mu \notin I_X, \quad \mu \neq j. \end{cases} \tag{5.22}$$

The vector $X(\theta)$ is a program of problem (5.1)–(5.3) if and only if

$$0 \leqslant \theta \leqslant \theta_0,$$

where θ_0 is the least of the three numbers

$$\theta_0^{(1)} = \min_{x_{ij} < 0} \frac{\alpha_{s_i} - x_{s_i}}{x_{ij}},$$

$$\theta_0^{(2)} = \min_{x_{ij} > 0} \frac{\beta_{s_i} - x_{s_i}}{x_{ij}},$$

$$\theta_0^{(3)} = x_j - \alpha_j = \beta_j - \alpha_j.$$

Let

$$\gamma_i = \begin{cases} \alpha_{s_i}, & \text{if } x_{ij} < 0, \\ \beta_{s_i}, & \text{if } x_{ij} > 0. \end{cases} \quad (i = 1, 2, \ldots, m+1). \tag{5.23}$$

In our notations, we have the following compact formula for θ_0:

$$\theta_0 = \min_{\substack{x_{ij} \neq 0 \\ 1 \leqslant i \leqslant m+1}} \frac{\gamma_i - x_{i0}}{x_{ij}}. \tag{5.24}$$

Here, as before, $x_{m+1;j} = -1$, $x_{m+1,0} = x_j = x_{s_{m+1}}$. For all nonnegative θ not larger than θ_0 (as defined in (5.24)), the elementary transformation associated with the vector A_j for which $x_j = \beta_j$ ($j \notin I_X$) transforms the support program X to a program $X(\theta)$ of problem (5.1)–(5.3). The proof of this proposition is identical with the proof of the analogous assertion for the first case, when $x_j = \alpha_j$, and can, therefore, be omitted.

In the following θ_0 will be called the elementary-transformation parameter.

Thus, (5.14) and (5.15) for $0 \leqslant \theta \leqslant \theta_0$, where θ_0 is computed, respectively, from (5.21) and (5.24), uniquely specify the elementary transformation associated with any restraint vector A_j, $j \notin I_X$.

Let us now trace the change of the linear form under the elementary transformation associated with the vector A_j, $j \notin I_X$.

Applying (5.14) (for the first case) and (5.22) (for the second case),

we obtain

$$L[X(\theta)]-L(X)=\begin{cases}\theta(c_j-\sum_{i=1}^{m}c_{s_i}x_{ij}) & \text{for } x_j=\alpha_j,\\ \theta(\sum_{i=1}^{m}c_{s_i}x_{ij}-c_j) & \text{for } x_j=\beta_j.\end{cases}$$

According to the definition of Δ_j (see (5.6) and (5.7)) we obtain the following result

$$L[X(\theta)]-L(X)=-\theta\Delta_j. \qquad (5.25)$$

Formula (5.25) shows that the parameter Δ_j can be interpreted as the evaluation of the vector A_j with respect to the program X with the given basis

$$A_{s_1},\ A_{s_2},\dots,A_{s_m}$$

or, more briefly, the *relative evaluation* of A_j.

5-5. We now give the description of the general procedure of solution of the linear-programming problem with bilateral restraints.

In this section problem (5.1)–(5.3) is assumed to be nondegenerate. In Chapter 2, 4-7, nondegeneracy was defined for the linear-programming problem in arbitrary form. It can easily be verified (see Exercise 5) that for a problem with bilateral restraints this definition can be given as follows.

A support program of problem (5.1)–(5.3) is called *nondegenerate*, if all its basis components satisfy inequalities (5.4).

In other words, a support program of a problem with bilateral restraints is considered nondegenerate if its m components satisfy restraints (5.4). As before, a *nondegenerate problem* is a problem whose support programs are all nondegenerate. To solve problem (5.1)–(5.3), we set out from some support program and the procedure is terminated after a finite number of homogenous iterations.

We will describe one step of the procedure. Let $X=(x_1, x_2,\dots,x_n)$ be a support program of problem (5.1)–(5.3) with the basis $A_{s_1}, A_{s_2},\dots,A_{s_m}$. The program X is given or has been obtained in the preceding iterations. Each iteration can be divided into two stages. In the first stage, the program X is tested for optimality. To this end the relative evaluations Δ_j of all the vectors A_j not appearing in the basis ($j\notin I_X$) are calculated. Depending on the method of computing z_j, the first or the second form of the optimality test is applied.

The following cases arise.

(a) If all the Δ_j (see (5.7)) prove to be nonnegative, the program X is optimal, and the solution procedure is terminated.

If the Δ_j of some of the restraint vectors $A_j (j\notin I_X)$ are negative we consider the elementary transformations associated with each of these vectors.

(b) Among these elementary transformations there is at least one whose θ_0 is infinite.

(c) For all the elementary transformations θ_0 is finite.

We analyze each of these cases.

From (5.21) and (5.24) we see that in case (a) there exists a vector A_k with a negative evaluation Δ_k such that

$$\begin{aligned}\alpha_{s_i}&=-\infty \text{ for all } x_{ik}>0,\\ \beta_{s_i}&=\ \ \infty \text{ for all } x_{ik}<0,\end{aligned}\qquad i=1,2,\dots,m+1,$$

if $x_k=\alpha_k$, and

$$\begin{aligned}\alpha_{s_i}&=-\infty \text{ for all } x_{ik}<0,\\ \beta_{s_i}&=\ \ \infty \text{ for all } x_{ik}>0,\end{aligned}\qquad i=1,2,\dots,m+1,$$

if $x_k=\beta_k$.

Obviously the vector $X(\theta)$ obtained under the elementary transformation associated with the vector A_k is a program of problem (5.1)–(5.3) for any $\theta>0$. Therefore, (5.25) (for $j=k$) shows that the linear form (5.1) is unbounded in the set of feasible programs of the problem. The solution procedure is terminated in this case when unsolvability of the problem is established. Observe that case (b) cannot arise if α_j and β_j are finite.

In case (c), we must proceed with the second stage of the iteration. In the second stage we construct a new support program for problem (5.1)–(5.3) which is closer to the optimum. The transition to the new program is effected by the elementary transformation of program X associated with any vector A_k whose evaluation Δ_k is negative. Let

$$X'=X(\theta_0),$$

where θ_0 is the parameter of the given elementary transformation. The components of the vector X' are determined from (5.14) if $x_k=\alpha_k$, and from (5.22) if $x_k=\beta_k$. The parameter θ_0 is computed from (5.21) or (5.24) depending on the actual limit with which the component x_k coincides. In all these relationships $j=k$.

For the sake of convention we say that θ_0 is obtained on the i-th position of the basis (or on the vector A_{s_i}) if

$$\theta_0=\begin{cases}\frac{x_{i0}-\gamma_i}{x_{ik}} & \text{for } x_k=\alpha_k,\\ \frac{\gamma_i-x_{i0}}{x_{ik}} & \text{for } x_k=\beta_k.\end{cases}$$

Here $i=1, 2, \ldots, m, m+1$.

We recall that the $(m+1)$-th position of the basis is occupied by the vector $A_k=A_{s_{m+1}}$ selected to be introduced into the basis of the new program.

Assume that θ_0 is obtained on the r-th position of the basis. Then, obviously, the component $x'_{s_r}=x_{s_r}(\theta_0)$ is equal to one of its limits (α_r or β_r). Thus, the components of the program X' with the indices $j\notin I_X$, $j\neq k$ and $j=s_r$ are equal to their limit values. It can easily be verified that the restraint vectors corresponding to the remaining variables are linearly independent. Indeed, for $r\neq m+1$ this follows from Theorem 2.1 of this chapter. If, however, $r=m+1$ (this case is quite feasible), the system of vectors being considered is the basis of the support program X and is a fortiori linearly independent.

We conclude that the new program X' is a support program of the problem. For $s_r=k$ the basis of program X' coincides with the basis of program X; for $s_r\neq k$ the basis of X' is obtained from the basis of X when the vector A_{s_r} is replaced by the vector A_k. The vector A_k is introduced into the r-th position of the basis.

It is worthy of note that since problem (5.1)–(5.3) is assumed nondegenerate, θ_0 can be obtained only on one position of the basis. Otherwise, the support program X' would be degenerate. Hence, the index r is in this case uniquely specified.

We see from (5.25) that when passing from support program X to support program X' the linear form increases

$$L(X')-L(X)=-\theta_0\Delta_k. \tag{5.26}$$

The elementary-transformation parameter θ_0 of a nondegenerate program is positive. Hence,

$$L(X')>L(X).$$

2154

The second stage of the iteration, and the iteration as a whole, is terminated by the construction of a support program X' with a higher value of the linear form.

5-6. Successive iterations are carried out as long as the required solution has not been obtained or unsolvability of the problem has not been established. In the nondegenerate case, each iteration increases the linear form (5.1). In the process of solving the problem we, therefore, continually deal with different support programs. It can easily be shown that the problem has at most $N=C_n^m \cdot 2^{n-m}$ support programs. Indeed, the number of bases of problem (5.1)–(5.3) is at most C_n^m, and to each basis correspond at most 2^{n-m} support programs (the extrabasis variables may equal either α_j or β_j). The process of solving a nondegenerate problem will, therefore, contain at most N iterations. In § 6 we shall show that in the degenerate case, too, the simplex method yields a solution of the problem or establishes its unsolvability after a finite number of steps.

In conclusion, we briefly repeat the sequence of operations involved in a single iteration. In the first stage of the iteration, the support program X is tested for optimality. The optimality test of a support program requires nonnegative Δ_j for all j. The relative evaluations Δ_j are expressed either in terms of the coefficients of the restraint vectors expressed in terms of the basis vectors or in terms of the components of the vector Λ.

The first stage leads to one of the three possibilities (cases (a), (b), (c)). In case (a), support program X is the solution of the problem. Case (b) proves that the linear form has no upper bound in the set of feasible programs of the problem. In both cases, the solution process is terminated. In case (c) we proceed with the second stage of the iteration. In this stage a new support program is constructed for which the value of the linear form is higher. The new program is construced with the aid of the elementary transformation associated with some vector A_k whose evaluation Δ_k with respect to the previous program is negative. We thus see that bilateral restraints almost do not affect the laboriousness of the computations of a single iteration in the simplex procedure.

§ 6. Degeneracy

6-1. Until now, in our discussion of the simplex method we invariably assumed the linear-programming problems to be nondegenerate. In the present section we extend the simplex method to the general case. The generalization is made for both nondegenerate and degenerate linear-programming problems. Since the canonical form of the general linear-programming problem is a particular case of a problem with bilateral restraints all the arguments (unless otherwise specified) refer to problem (5.1)–(5.3).

The characteristic feature of a degenerate support program of problem (5.1)–(5.3) is that some of its basis components obtain their limiting values (α_j or β_j). Any nondegenerate support program has a unique basis comprising the restraint vectors whose components satisfy inequalities (5.4). In the degenerate case, any given support program may have several bases. For example, if a support program of problem (1.1)–(1.3) has $\nu<m$ positive components, its basis is made up of ν restraint vectors corresponding to these components and $m-\nu$ restraint vectors associated with the zero components of the program. The only requirement needed to be fulfilled by the $m-\nu$ vectors is that the resulting system of m restraint vectors be linearly independent, Obviously, the number of different bases of such a support program of problem (1.1)–(1.3) is at most $C_{n-\nu}^{m-\nu}$, and in some problems this maximum estimate is actually obtained. Analogous considerations apply to problems with bilateral restraints (see Exercise 6).

Below we examine all the stages of a single iteration of the simplex method stressing those points in which nondegneracy is assumed.

Thus, let

$$X=(x_1, x_2, \ldots, x_n)$$

be a support program of problem (5.1)–(5.3) with the basis $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$. The first stage of the iteration does not imply nondegeneracy of program X. In the second stage of the iteration the assumption of nondegeneracy is used twice:

(1) when computing θ_0;

(2) when choosing the vector to be eliminated from the basis (or, equivalently, when determining the basis position in which a new vector is to be introduced).

Let us consider each of these cases separately:

1. When program X is degenerate, θ_0 may equal zero if the component of the vector A_k introduced into the basis is equal to α_k, which applies when for some i, $1 \leqslant i \leqslant m$,

$$x_{i0}=x_{s_i}=\alpha_{s_i}, \quad x_{ik}>0,$$

or

$$x_{i0}=x_{s_i}=\beta_{s_i}, \quad x_{ik}<0$$

(see (5.21)).

When $x_k=\beta_k$ $\theta_0=0$ if for some i

$$x_{i0}=x_{s_i}=\alpha_{s_i}, \quad x_{ik}<0,$$

or

$$x_{i0}=x_{s_i}=\beta_{s_i}, \quad x_{ik}>0$$

(see (5.24)).

If $\theta_0=0$, the new support program X' coincides with the previous support program X. The iteration only modifies the basis of this program.

2. The support program X' of the nondegenerate problem (5.1)–(5.3) which is obtained when program X is improved is nondegenerate.

Hence, θ_0 is obtained on a single restraint vector A_{s_r} (on a single basis position, r). The single vector A_{s_r} is eliminated from the basis. In the general case, however, θ_0 may be obtained on several vectors. Only one of these vectors should be eliminated from the basis.

If in several successive iterations $\theta_0=0$, the solution procedure during these iterations involves transferring to different bases of the same support program. Since the linear form obviously remains constant in the process, there is no justification of the assertion that different bases of the problem are involved.

In Chapter 5, § 9, we shall examine a linear-programming problem whose solution entails a so-called c y c l e, i.e., periodic return, after several iterations, to a given basis. The solution procedure of such a problem will, obviously, not terminate. We shall show, however, that a slight improvement of the procedure will entirely eliminate the possibility of a cycle arising. It will be proved that if the vector to be eliminated from the basis is chosen, from among the several vectors on which θ_0 is obtained, in a particular way, we invariably deal with different bases of the problem and, therefore, after a finite number of iterations obtain an optimal program.

6-2. Before giving the rule for vector elimination, we shall illustrate degeneracy geometrically.

We shall apply the first geometrical interpretation. Consider a support program $\bar{X}=(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$ of a problem with bilateral restraints (5.1)–(5.3). To be specific, we assume that the first ν components, $\nu \leqslant m$, of this program satisfy condition (5.4). The support program $\bar{X}$ corresponds to the vertex $\bar{X}$ of the polyhedral restraint set. The vertex $\bar{X}$ is the intersection of $m+n-\nu$ hyperplanes the first m of which are defined by restraints (5.2) and the other $n-\nu$ by the equations

$$x_j=\bar{x}_j=\bar{\gamma}_j, \qquad j=\nu+1, \ldots, n, \tag{6.1}$$

where $\bar{\gamma}_j$ is either α_j or β_j.

If $\nu=m$, the vertex $\bar{X}$ belongs to exactly n hyperplanes. If $\nu<m$, the number of hyperplanes intersecting at the vertex is greater than the dimensionality n of the space.

Geometrically, degeneracy of a support program implies that the number of hyperplanes intersecting at the corresponding vertex is greater than n, where n is the number of problem variables.

Consider a basis of program $\bar{X}$,

$$A_1, A_2, \ldots, A_\nu, A_{j_1}, A_{j_2}, \ldots, A_{j_{m-\nu}},$$

where $\nu+1 \leqslant j_s \leqslant n$, $s=1, 2, \ldots, m-\nu$. Geometrically, the choice of this basis indicates that the vertex corresponding to program $\bar{X}$ is considered as the intersection of exactly n independent hyperplanes, the first m of which are defined, as always, by restraints (5.2), and the remaining $n-m$ are hyperplanes of the form (6.1) with $j \neq j_s$, $s=1, 2, \ldots, m-\nu$. For $\nu<m$ there may exist several such systems of hyperplanes and, consequently, several bases of program $\bar{X}$.

Let A_j be an arbitrary restraint vector not appearing in the given basis of program $\bar{X}$. We eliminate the hyperplane $x_j=\bar{y}_j$ from the system of hyperplanes associated with the basis. The intersection of the remaining hyperplanes is a line passing through point $\bar{X}$. We shall say that the ray issuing from the point $\bar{X}$ parallel to this line toward the polyhedral set of the problem *corresponds* to the vector A_j. Analogous definition for the canonical form of the linear-programming problem will be found in § 4.

The elementary transformation of program $\bar{X}$ corresponding to the vector A_j is geometrically equivalent to translation along this ray. While in the nondegenerate case the intersection of any such ray with the polyhedral restraint set is, necessarily, a segment, this intersection in the degenerate case may contain only one point, $\bar{X}$. Thus, any infinitesimal translation along the given ray gives rise to a point which does not belong to the domain of definition of the problem.

Let the vector A_k be chosen to be introduced into the basis in the first stage of an iteration. The linear form of the problem increases along the ray corresponding to this vector. If by translation along this ray we do not leave the polyhedral restraint set ($\theta_0>0$), the second stage will give a new support program on which the value of the linear form is higher. If no such translation is possible ($\theta_0=0$), we remain at the same vertex $\bar{X}$. The second stage of the iteration will only modify the basis of program $\bar{X}$. One of the hyperplanes $x_k=\bar{x}_k$ in the set of hyperplanes corresponding to the previous basis will be replaced by a hyperplane of the form (6.1) with $j=j_s$, $s=1, 2, \ldots, m-\nu$. If we do not specify which of these hyperplanes should in fact be introduced in the system corresponding to the new basis, we may arrive after several iterations to a system of hyperplanes which has already been examined, i. e., a cycle arises.

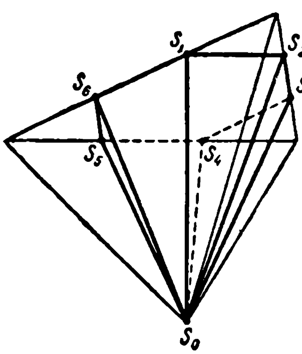


FIGURE 4.7

To clarify the geometrical meaning of degeneracy, consider a linear-programming problem whose restraint polyhedron is a regular hexahedral pyramid (Figure 4.7) (the problem restraints are assumed to be defined by inequalities).

The vertex S_0 of the restraint polyhedron correspond to a degenerate support program. Consider the six planes to which the faces intersecting at S_0 belong. Any three of these planes represent some basis of the support program defined by S_0. The support program in question thus has $C_6^3=20$ different bases.

For example, consider the basis corresponding to the faces $S_1S_0S_2$, $S_3S_0S_4$, $S_5S_0S_6$. The lines of intersection of any two planes to which these faces belong have only one point in common with the restraint polyhedron (the point S_0). Therefore, any single iteration will invariably leave us at point S_0, i.e., we shall pass from the given basis of the program to another basis of the same program. This also applies to the basis comprising the faces $S_2S_0S_3$, $S_4S_0S_5$, $S_6S_0S_1$.

Any other three of the six planes in question obviously define at least one direction which has a whole segment in common with the restraint polyhedron. On the other hand, in any such threesome there are two planes whose line of intersection has only one point in common with the polyhedron. From any such basis we may, therefore, proceed, after one iteration, to a basis of a new support program, or to a different basis of the same program.

6-3. We now consider the concept of degeneracy in terms of the second geometrical interpretation (see Chapter 2, § 5, and Chapter 4, § 4). Since the second geometrical interpretation has been described only for linear-programming problems in canonical form, we shall limit the discussion to problem (1.1)–(1.3).

Let

$$\bar{X}=(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$$

be a support program of problem (1.1)–(1.3). To be specific, we assume that this basis comprises the first m restraint vectors. Then,

$$\bar{x}_{m+1}=\ldots=\bar{x}_n=0.$$

Let the first ν basis components of program $\bar{X}$ be positive, and the remaining components equal zero.

For $\nu<m$ the support program $\bar{X}$ is said to be degenerate. Program $\bar{X}$ corresponds to some point P on the intersection of line Q and cone K of the given problem. This point also belongs to the m-dimensional cone $K_{\bar{X}}$ spanned by the augmented basis vectors of program $\bar{X}$. If $\nu<m$, the point P is on the boundary of cone $K_{\bar{X}}$ (more precisely, inside its ν-dimensional face, which is a cone spanned by the first ν augmented restraint vectors).

Let the vector A_k be chosen to be introduced into the basis. We have seen (§ 4) that the new support program corresponds to the highest point of intersection of the line Q with the cone K_k spanned by the vectors

$\bar{A}_1, \bar{A}_2, \ldots, \bar{A}_m, \bar{A}_k$. If program $\bar{X}$ is nondegenerate ($\nu = m$), the intersection of line Q and cone K_k is a segment (the problem is assumed to be solvable). The highest point of intersection of Q and K_k is therefore distinct from the lowest point P. If, however, program $\bar{X}$ is degenerate, the line Q may intersect the cone K_k in one point (the point P), which is on the boundary of the cone $K_{\bar{X}}$, i.e., one of the faces of the cone K_k. In this case the highest point of intersection coincides with the lowest point P, and we have the same support program $\bar{X}$.

For $m=2$ all the preceding geometrical considerations become clear. Consider a linear-programming problem described geometrically in Figure 4.8. The support program with basis A_1, A_5 is obviously degenerate, since line Q intersects the side $O\bar{A}_5$ of the angle $\bar{A}_1 O \bar{A}_5$. Introducing the vector A_4 into the basis does not increase the linear form, since the line Q and the cone spanned by the vectors $\bar{A}_1, \bar{A}_5, \bar{A}_4$ intersect at one point, P. As regards the vector A_3, its introduction into the basis increases the linear form (the intersection of the line Q and the cone spanned by the vectors $\bar{A}_1, \bar{A}_5, \bar{A}_3$ is the segment PP').

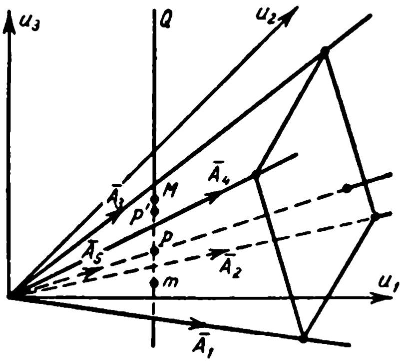


FIGURE 4.8

This geometrical interpretation of degeneracy points to the way of avoiding the danger of a cycle. Degeneracy of a problem means that the line Q intersects at least one of the polyhedral cones spanned by at most $m-1$ augmented restraint vectors. It is obvious, geometrically, that some small parallel translation of the line Q will eliminate all these intersections, i.e., will reduce the problem to a nondegenerate one in which no cycle may arise. Translation of the line Q is associated with modification of the constraint vector B which uniquely defines the position of the line. A cycle can thus be avoided if the vector B is suitably modified. This idea forms the basis of the rule derived for determining the vector to be eliminated from the basis without obtaining a cycle.

6-4. This rule, which is derived in what follows, was first obtained by Charnes /117/ for the linear-programming problem in canonical form.

Let $\varepsilon > 0$ and $R_1, R_2, \ldots, R_m$ be a linearly independent system of m-dimensional vectors. Consider the following linear-programming problem:

Maximize the linear form

$$\sum_{j=1}^{n} c_j x_j \tag{6.2}$$

subject to the conditions

$$\sum_{j=1}^{n} x_j A_j = B(\varepsilon), \tag{6.3}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j, \quad j = 1, 2, \ldots, n. \tag{6.4}$$

Here

$$B(\varepsilon) = B + \sum_{j=1}^{m} \varepsilon^j R_j. \tag{6.5}$$

Problem (6.2)–(6.4) differs from problem (5.1)–(5.3) only in the constraint vector, which is a combination of the vector B, the system $R_1, R_2, \ldots, R_m$ and the number ε.

Each linear-programming problem (5.1)–(5.3) is thus associated with an entire set of problems (6.2)–(6.4). The representatives of this set will be called ε-problems.

Our following analysis will proceed from two propositions on the ε-problems.

Theorem 6.1. There exists an $\varepsilon_1 > 0$ such that the ε-problem (6.2)–(6.4) for $0 < \varepsilon < \varepsilon_1$ is nondegenerate.

Proof. Let $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$ be a linearly independent system of restraint vectors. The set of indices of these vectors is denoted by I. We associate with the system of vectors A_j, $j \in I$, a set of n-dimensional vectors $X(\varepsilon)$ satisfying restraints (6.3) and such that $x_j(\varepsilon)$ for $j \notin I$ is equal either to α_j of to β_j.

From (6.3) and (6.5)

$$\sum_{j \in I} x_j(\varepsilon) A_j = B + \sum_{j=1}^{m} \varepsilon^j R_j - \sum_{j \notin I} \gamma_j A_j,$$

where $\gamma_j = x_j(\varepsilon)$ is either α_j or β_j. Hence

$$x_{s_i}(\varepsilon) = \bar{b}_i - \sum_{j \notin I} \gamma_j \bar{a}_{ij} + \sum_{j=1}^{m} \varepsilon^j \bar{r}_{ij}. \tag{6.6}$$

Here $\bar{b}_i$, $\bar{a}_{ij}$, $\bar{r}_{ij}$ are the coefficients of A_{s_i} when the vectors B, A_j and R_j, respectively, are expressed in terms of the vectors of system A_j, $j \in I$. Note that none of the polynomials $x_{s_i}(\varepsilon)$ is constant. Indeed, the square matrix $\|\bar{r}_{ij}\|_m$ is the product of the matrices $(A_{s_1}, A_{s_2}, \ldots, A_{s_m})^{-1}$ and $\|r_{ij}\|_m = (R_1, R_2, \ldots, R_m)$, each of which has a nonzero determinant. The determinant of the matrix $\|\bar{r}_{ij}\|_m$ is also nonzero and, consequently, none of the rows of this matrix can comprise zeros only.

Thus, for any i, some of the $\bar{r}_{ij}$, $j = 1, 2, \ldots, m$ are nonzero, i.e., the polynomials $x_{s_i}(\varepsilon) \not\equiv \text{const}$. Hence, the polynomials $x_{s_i}(\varepsilon) - \alpha_{s_i}$ and $\beta_{s_i} - x_{s_i}(\varepsilon)$ do not vanish identically. Any polynomial (which does not vanish identically) of degree not greater than m has at most m positive roots.

Let the least of the positive roots of the polynomials $x_{s_i}(\varepsilon) - \alpha_{s_i}$ and $\beta_{s_i} - x_{s_i}(\varepsilon)$ be $\eta_i(I, \gamma_j)$. If there are no positive roots, we take $\eta_i = \infty$. Let the least $\eta_i(I, \gamma_j)$, $i = 1, 2, \ldots, m$ be $\eta(I, \gamma_j)$. The parameter $\eta(I, \gamma_j)$ is defined by a system of linearly independent vectors A_j, $j \in I$, and by the number γ_j, $j \notin I$. The n restraint vectors can be arranged into a finite number of different systems each comprising m vectors. For fixed I, there exist at most 2^{n-m} different sets of γ_j, $j \notin I$. Therefore,

$$\varepsilon_1 = \min \eta(I, \gamma_j),$$

where the minimum taken over all the possible systems of linearly independent vectors A_j, $j \in I$ and sets of γ_j, equal to α_j or β_j, $j \notin I$, is positive.

Now let $0 < \varepsilon < \varepsilon_1$. Consider a support program of the ε-problem with the basis $A_{s_1}, \ldots, A_{s_m}$. It is obvious that the program is nondegenerate. Indeed, if for some i the basis component $x_{s_i}(\varepsilon)$ is equal to α_{s_i} or β_{s_i}, one of the polynomials

$$x_{s_i}(\varepsilon) - \alpha_{s_i}, \quad \beta_{s_i} - x_{s_i}(\varepsilon)$$

has a root ε.

On the other hand, according to the definition of ε_1, the positive roots of these polynomials cannot be smaller than $\varepsilon_1 > \varepsilon$.

Thus, the basis components of any support program of the ε-problem for $0 < \varepsilon < \varepsilon_1$ lie strictly between their boundary values which, by definition, implies nondegeneracy of problem (6.2)–(6.4). This completes the proof.

Theorem 6.2. There exists an $\varepsilon_2 > 0$ such that if

$$0 < \varepsilon < \varepsilon_2$$

and $X(\varepsilon) = (x_1(\varepsilon), \ldots, x_n(\varepsilon))$ is a support program of the ε-problem with the basis $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$, the vector $X = (x_1, x_2, \ldots, x_n)$, where $x_j = x_j(\varepsilon)$ for $j \neq s_i$, $i = 1, 2, \ldots, m$ and the other m components are determined from equations (5.2), is a support program of problem (5.1)–(5.3). Moreover, if $X(\varepsilon)$ is an optimal program of the ε-problem, program X is a solution of problem (5.1)–(5.3).

Proof. Consider a system of m linearly independent restraint vectors A_j, $j \in I = (s_1, s_2, \ldots, s_m)$. Let the n-dimensional vector X satisfy restraints (5.2). Let, moreover, $x_j = \gamma_j$ for $j \notin I$ be equal to α_j or β_j. Then, from (6.6),

$$x_{s_i} = \bar{b}_i - \sum_{j \notin I} \gamma_j \bar{a}_{ij}, \qquad i = 1, 2, \ldots, m. \tag{6.7}$$

Let the least of the positive numbers $x_{s_i} - \beta_{s_i}$ and $\alpha_{s_i} - x_{s_i}$ be $\xi_i(I, \gamma_j)$. If $\alpha_{s_i} \leq x_{s_i} \leq \beta_{s_i}$, we take $\xi_i(I, \gamma_j) = \infty$. Let

$$\xi(I, \gamma_j) = \min_{1 \leq i \leq m} \xi_i(I, \gamma_j).$$

The parameter $\xi(I, \gamma_j)$ is specified by the system of vectors A_j, $j \in I$, and by the set of numbers γ_j, $j \notin I$, each of which may assume at most two values (α_j, β_j). Therefore,

$$\xi = \min_{I, \gamma_j} \xi(I, \gamma_j),$$

as the least of a finite set of positive numbers is also positive.

We now introduce a number $r(I)$ equal to the modulus of the greatest (in absolute value) element of the matrix $\|\bar{r}_{ij}\|_m$ constructed of the coefficients of the vectors $R_1, R_2, \ldots, R_m$ expressed in terms of the vectors of the system A_j, $j \in I$. Let

$$r = \max_I r(I).$$

Obviously, $r>0$. We take

$$\varepsilon_2'=\min\left(\frac{\xi}{rm},\ 1\right)>0.$$

We shall show that for any ε, $0<\varepsilon<\varepsilon_2'$, to each support program of the ε-problem corresponds a support program of problem (5.1)–(5.3) having the same basis and the same extrabasis variables.

Let $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$ be the basis of the support program $X(\varepsilon)=(x_1(\varepsilon), \ldots, x_n(\varepsilon))$ of the ε-problem. The vector $X=(x_1, x_2, \ldots, x_n)$ satisfies equalities (5.2), and $x_j=x_j(\varepsilon)=\gamma_j$ for $j\notin I$. The basis components of program $X(\varepsilon)$, which are defined according to (6.6) by the equalities

$$x_{s_i}(\varepsilon)=\bar{b}_i-\sum_{j\notin I}\gamma_j\bar{a}_{ij}+\sum_{j=1}^{m}\varepsilon^j\bar{r}_{ij}=x_{s_i}+\sum_{j=1}^{m}\varepsilon^j\bar{r}_{ij},\qquad i=1, 2, \ldots, m, \tag{6.8}$$

satisfy the conditions

$$\alpha_{s_i}\leqslant x_{s_i}(\varepsilon)\leqslant\beta_{s_i}.$$

We shall verify that

$$\alpha_{s_i}\leqslant x_{s_i}=\bar{b}_i-\sum_{j\notin I}\gamma_j\bar{a}_{ij}\leqslant\beta_{s_i},\qquad i=1, 2, \ldots, m. \tag{6.9}$$

Indeed, if for some $i=k$ the component x_{s_k} is, e.g., to the left of α_{s_k}, we have

$$x_{s_k}(\varepsilon)=x_{s_k}+\sum_{j=1}^{m}\varepsilon^j\bar{r}_{kj}\leqslant\alpha_{s_k}-\xi+\sum_{j=1}^{m}\varepsilon^j\bar{r}_{kj}\leqslant\alpha_{s_k}-\xi+r\sum_{j=1}^{m}\varepsilon^j\leqslant$$
$$\leqslant\alpha_{s_k}-\xi+mr\varepsilon<\alpha_{s_k}.$$

The first and the second inequalities follow from the definition of ξ and r, respectively. The third and the fourth inequalities follow from the definition of $\varepsilon_2'>\varepsilon$.

The above relationships yield the inequality

$$x_{s_k}(\varepsilon)<\alpha_{s_k},$$

which contradicts the assumption that $x_{i_k}(\varepsilon)$ is a component of a program of the ε-problem. Thus,

$$x_{s_i}\geqslant\alpha_{s_i},\qquad i=1, 2, \ldots, m.$$

The restraints in the right-hand side of (6.9) are analogously verified.

The vector X thus satisfies restraint (5.2) and restraints (5.3) simultaneously and is, consequently, a program of problem (5.1)–(5.3). By definition, the components x_j of X for $j\notin I$ are equal to their boundary values. Therefore X is a support program of problem (5.1)–(5.3) with the basis $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$.

Let now $X(\varepsilon)$ be an optimal program of the ε-problem. Let further $0<\varepsilon<\varepsilon_1$. From Theorem 6.1, the ε-problem is nondegenerate. Hence the vector $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$ determined from

$$(\Lambda, A_j)=c_j,\quad j\in I, \tag{6.10}$$

satisfies the inequality

$$(\Lambda, A_j)\geqslant c_j, \tag{6.11}$$

if $j\notin I$ and $x_j(\varepsilon)=x_j=\alpha_j$ or the inequality

$$(\Lambda, A_j)\leqslant c_j, \tag{6.11'}$$

if $j\notin I$ and $x_j(\varepsilon)=x_j=\beta_j$.

Indeed, if (6.11) and (6.11') did not apply, we could have increased, by a single simplex iteration, the value of the linear form of the nondegenerate ε-problem.

Since the support program X and $X(\varepsilon)$ have the same basis, relationships (6.10), (6.11), and (6.11') can be considered, with reference to program X, as the conditions of the second form of the optimality test. Program X therefore solves problem (5.1)–(5.3).

The numbers ε_2 entering the conditions of Theorem 6.2 can thus be defined as

$$\varepsilon_2=\min\left\{\varepsilon_2',\ \varepsilon_1\right\}>0.$$

This completes the proof.

6-5. These properties of the set of ε-problems associated with problem (5.1)–(5.2) suggest the following procedure for solving the problem without introducing, in any way, the requirement of nondegeneracy. The initial problem (5.1)–(5.3) is replaced by the corresponding ε-problem with $0<\varepsilon<\varepsilon_0$, where

$$\varepsilon_0=\min\{\varepsilon_1,\ \varepsilon_2\}=\varepsilon_2.$$

Let the set of feasible programs of this ε-problem be nonempty. From Theorem 6.1 ($0<\varepsilon<\varepsilon_1$), this ε-problem is nondegenerate. The simplex method will, therefore, produce, after a finite number of iterations, its support solution.

From Theorem 6.2 ($0<\varepsilon<\varepsilon_2$) the basis of the solution is also the basis of the optimal program of problem (5.1)–(5.3). The extrabasis components of this program are equal to the corresponding components of the solution of the ε-problem. Hence, to obtain the basis components of the optimal program required it suffices to express the vector

$$B - \sum_{j \notin I^*} x_j^*(\varepsilon) A_j$$

in terms of the system A_j, $j \in I^*$. Here I^* is the set of indices of restraint vectors which constitute the basis of the solution of the ε-problem:

$$X^*(\varepsilon) = (x_1^*(\varepsilon), x_2^*(\varepsilon), \ldots, x_n^*(\varepsilon)).$$

The ε-problem in question was assumed to be solvable. Note that if the process of solution of ε-problem terminates in case (b), indicating its unsolvability, unsolvability of the principal problem (5.1)–(5.3) is also implied. The proof of this proposition follows from Exercise 7. To apply the ε-procedure, the number ε_0 need not be determined; only its existence need be established.

The solution of the ε-problem can be attempted with any sufficiently small ε. To determine a solution of problem (5.1)–(5.3) it suffices to take $\varepsilon=0$ in the optimal program of the ε-problem.

Since to each support program of the ε-problem with $0<\varepsilon<\varepsilon_2$ there corresponds a support program of problem (5.1)–(5.3) with the same basis (Theorem 6.2), the solution procedure of the ε-problem involves examination of support programs of problem (5.1)–(5.3). In a finite number of iterations the solution is obtained. We now establish the rules by which such examinations are made possible without considering the set of ε-problems.

Let $X=(x_1, x_2, \ldots, x_n)$ be a support program of problem (5.1)–(5.3) with the basis $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$. Let the vector $X(\varepsilon)$, where $x_j(\varepsilon)=x_j$ for $j \notin I_X$, be a program of the ε-problem for $0<\varepsilon<\varepsilon_0$.

If the simplex method is now applied to the ε-problem, starting with the support program $X(\varepsilon)$, then in a finite number of iterations, having examined the programs $X^{(1)}(\varepsilon)$, $X^{(2)}(\varepsilon)$, ..., $X^{(N)}(\varepsilon)$, we obtain the solution $X^*(\varepsilon)$. The same procedure can be carried out by examining the support program $X^{(t)}(0)=X^{(t)}$, $t = 1, 2, \ldots, N$ of problem (5.1)–(5.3) and hereby the solution $X^*(0)=X^*$ is reached.

We shall illustrate the above by transformation from program X to program $X^{(1)}$. By assumption, the bases of programs X and $X(\varepsilon)$ and their extrabasis components are identical. The first stage of the iteration can, therefore, be realized starting with program X. In the analysis of the second stage, as in the case of problems with bilateral restraints, we shall distinguish between two possibilities.

1. The component associated with the vector A_k chosen to be introduced into the new basis assumes its left boundary value α_k. To determine the vector to be eliminated from the basis, we compute, from (5.21),

$$\theta_0(\varepsilon) = \min_{\substack{x_{ik} \neq 0 \\ 1 \leq i \leq m+1}} \frac{x_{i0}(\varepsilon) - \gamma_i}{x_{ik}}. \tag{6.12}$$

Inserting for $x_{i0}(\varepsilon) = x_{s_i}(\varepsilon)$, $i=1, \ldots, m+1$ in (6.12) its expression from (6.8), we have

$$\theta_0(\varepsilon) = \min_{\substack{x_{ik} \neq 0 \\ 1 \leq i \leq m+1}} \frac{x_{i0} + \sum_{j=1}^{m} \varepsilon^j \bar{r}_{ij} - \gamma_i}{x_{ik}}. \tag{6.13}$$

Here, as before, $\bar{r}_{ij}$, $i=1, 2, \ldots, m$ are the coefficients of the vector R_j, $j=1, 2, \ldots, m$, expressed in terms of the system $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$ constituting the basis of programs X and $X(\varepsilon)$. Note that $\bar{r}_{m+1,j}=0$, $j=1, 2, \ldots, m$, since $x_{m+1,0}(\varepsilon)=x_k(\varepsilon)=x_k$.

Since the ε-problem is nondegenerate, there exists a unique index r on which $\theta_0(\varepsilon)$ is obtained. The vector A_k is introduced in the r-th position of the basis to replace the original vector A_{s_r}.

Relationship (6.13) shows that the index r, for sufficiently small ε, can be obtained in the following way. A sequence of sets of indices i are determined recurrently:

$$E_0 \supseteq E_1 \supseteq \ldots \supseteq E_t.$$

Let E_0 be the set of indices i on which

$$\theta_0 = \min_{\substack{x_{ik} \neq 0 \\ 1 \leq i \leq m+1}} \frac{x_{i0} - \gamma_i}{x_{ik}} \tag{6.14}$$

is obtained. If the set E_d contains more than one element, we introduce the set E_{d+1} comprising the indices $i \in E_d$ on which

$$\min_{i \in E_d} \frac{\bar{r}_{i,d+1}}{x_{ik}} \tag{6.15}$$

is obtained. Otherwise, the process is terminated. Thus, the sequence of sets E_d ends when a set E_t containing one element is obtained. It follows from (6.13) that, $\varepsilon>0$ being sufficiently small, this sequence produces the index r on which $\theta_0(\varepsilon)$ is obtained. Since the ε-problem is nondegenerate, $\theta_0(\varepsilon)$ is obtained on a unique index and therefore, having formed a sequence of at most $m+1$ sets E_d, we obtain a set E_t containing only one element r, the index of the basis position into which the vector A_k is to be introduced.

2. The component associated with the vector A_k chosen to be introduced into the new basis assumes its right boundary value β_k. To determine the vector to be eliminated from the basis, we must compute (see (5.24))

$$\theta_0(\varepsilon)= \min_{\substack{x_{ik}\neq 0\\ 1\leqslant i\leqslant m+1}} \frac{\gamma_i - x_{i0}(\varepsilon)}{x_{ik}}. \tag{6.16}$$

Relationship (6.16) can be rewritten in the form

$$\theta_0(\varepsilon)= \min_{\substack{x_{ik}\neq 0\\ 1\leqslant i\leqslant m+1}} \frac{\gamma_i - x_{i0} - \sum_{j=1}^{m} \varepsilon^j \bar{r}_{ij}}{x_{ik}}. \tag{6.17}$$

Hence, as in the first case, it follows directly that the index r of the basis position into which the vector A_k should be introduced can be determined as follows. We form a set E_0 of the indices i on which

$$\theta_0= \min_{\substack{x_{ik}\neq 0\\ 1\leqslant i\leqslant m+1}} \frac{\gamma_i - x_{i0}}{x_{ik}} \tag{6.18}$$

is obtained. The sequence $\{E_d\}$ is then recurrently constructed:

$$E_0 \supseteq E_1 \supseteq \ldots \supseteq E_t.$$

If E_d is found to contain more than one element, the set E_{d+1} is defined as containing the indices $i \in E_d$ on which

$$\max_{i \in E_d} \frac{\bar{r}_{i,d+1}}{x_{ik}} \tag{6.19}$$

is obtained. If E_d contains one element, the sequence is terminated ($E_d = E_t$). The sequence of sets E_d containing at most $m+1$ sets terminates, necessarily, with the set E_t containing one element r, the index of the required basis position.

The preceding rule for establishing the vector to be eliminated from the basis ensures that no cycle will arise when solving problem (5.1)–(5.3). This rule is, thus, always suitable for obtaining the solution of problem (5.1)–(5.3) in a finite number of simplex iterations.

We should again emphasize that to realize the rule we must

(1) isolate a linearly independent system of vectors $R_1, R_2, \ldots, R_m$ such that for a sufficiently small $\varepsilon > 0$ the vector $X(\varepsilon)$ corresponding to the initial program X is a program of the ε-problem;

(2) know the coefficients of the vectors $R_1, R_2, \ldots, R_m$ as expressed in terms of any current basis.

When describing the simplex algorithms we shall always indicate the most expedient (from the computational point of view) system of vectors $R_1, R_2, \ldots, R_m$.

6-6. The canonical form of the linear-programming problem is a particular case of a problem with bilateral restraints ($\alpha_j=0$, $\beta_j=\infty$). Therefore, for problem (1.1)–(1.3), the selection rule for basis vector elimination is a particular case of the general rule. We now formulate this particular rule.

The set E_0 of the indices i on which

$$\theta_0= \min_{\substack{x_{ik} > 0\\ 1\leqslant i\leqslant m}} \frac{x_{i0}}{x_{ik}}$$

is obtained is constructed. If E_0 contains one element r, the vector A_k is introduced into the r-th basis position. Otherwise, the set E_1 is constructed of the indices $i \in E_0$ on which

$$\min_{i \in E_0} \frac{\bar{r}_{i1}}{x_{ik}}$$

is obtained. If E_1 contains one element r, we know which vector A_{i_r} should be eliminated from the basis. Otherwise, the set E_2 is constructed, etc. Finally we obtain, necessarily, a set E_t, $0 < t \leq m$, containing one element, which specifies the vector to be eliminated from the basis.

§ 7. Methods for construction of an initial program

7-1. In our discussion in preceding sections we proceeded from some initial support program without specifying in any way how this support program is to be obtained. There are various classes of problems where the determination of an initial support program is quite trivial.

Let, for example, the restraints of a linear-programming problem be given in the form

$$\sum_{j=1}^{n} A_j x_j \leq B, \quad x_j \geq 0,$$

and the components of the vector B be nonnegative. In this case the problem is reduced to the canonical form by introducing m additional nonnegative variables $x_{n+1}, \ldots, x_{n+m}$ and corresponding unit restraint vectors $A_{n+1}, \ldots, A_{n+m}$:

$$A_{n+i} = (\underbrace{\overbrace{0, \ldots, 0, 1}^{i}, 0, \ldots, 0}_{m})^T.$$

In the linear form of the problem, the coefficients of the additional variables are all zero.

An initial support program of the problem in question with its restraints reduced to canonical form is, obviously,

$$X = (\underbrace{0, \ldots, 0}_{n}, b_1, \ldots, b_m).$$

The basis of this initial support program comprises the unit vectors A_{n+i} ($i = 1, \ldots, m$). The components x_{ij} in the expression of A_j in terms of the basis vectors A_{n+i}, $i = 1, \ldots, m$ are equal to the corresponding components of the restraint vector A_j.

Since there is no need to solve systems of linear equations to determine the x_{ij} in the initial stage, the corresponding computations of the simplex procedure are substantially simplified. However, it must be kept in mind that if we take the additional vectors as the initial basis vectors we, usually, start with a poorer first approximation than in cases when the basis contains at the outset some vectors corresponding to the principal variables. As a rule, there are more iterations required for computing the optimal program by this technique. Nevertheless, since modern computers are used, it is justified to introduce the additional variables into the basis. The increased number of computations, which may occur in this case, only involves repeated similar operations, which are quite simple (see Chapter 5, §§ 2, 5).

There are other classes of problems for which determination of an initial support program requires substantially less computations than solution of a problem with a given initial program. Unfortunately, this is not

always so with the general linear-programming problem. In general, computation of an initial support program is no less tedious than determination of a solution when the initial support program is given.

We now give some general methods for computation of initial support programs. In the following (Chapter 5, § 8) we shall return to this problem and indicate several instances for which these methods can be simplified.

7-2. Consider a linear-programming problem in canonical form:

Maximize the linear form

$$L(X)=\sum_{j=1}^{n} c_j x_j \tag{7.1}$$

subject to the conditions

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \qquad i=1, 2, \ldots, m, \tag{7.2}$$

$$x_j \geqslant 0, \qquad j=1, 2, \ldots, n. \tag{7.3}$$

Here we may take all $b_i \geqslant 0$. Otherwise, all the signs should be reversed in both sides of the corresponding equality.

Also consider the following auxiliary problem:

Maximize the linear form

$$\tilde{L}(X)=-\sum_{i=1}^{m} x_{n+i} \tag{7.4}$$

subject to the conditions

$$\sum_{j=1}^{n} a_{ij}x_j + x_{n+i} = b_i, \qquad i=1, 2, \ldots, m, \tag{7.5}$$

$$x_j \geqslant 0, \qquad j=1, 2, \ldots, n+m. \tag{7.6}$$

We solve the auxiliary problem by the simplex method. A feasible initial program of the auxiliary problem is made up of the components

$$x_j = 0 \quad \text{for } j \leqslant n,$$
$$x_{n+i} = b_i \quad \text{for } i=1, 2, \ldots, m.$$

To the components x_j of the initial program with $j \geqslant n+1$ correspond m different unit restraint vectors. This indicates that the initial program chosen is a support program of the auxiliary problem. The linear form $\tilde{L}$ is bounded above in the set of feasible programs of the auxiliary problem ($\tilde{L} \leqslant 0$). Therefore, by the simplex method the optimal support program of the problem will be obtained after a finite number of iterations. Two cases are possible:

a) The optimal value $\tilde{L}^*$ is zero.

b) The optimal value $\tilde{L}^*$ is negative.

In case (a) the optimal program of the auxiliary problem is a support program of the initial problem (7.1)-(7.3). Indeed, when $\tilde{L}^*=0$, all $x^*_{n+i}=0$ $(i=1, 2, \ldots, m)$. Hence, the solution X^* (its first n components) of the auxiliary problem satisfies the restraints of the initial problem. The optimal program of the auxiliary problem thus coincides with some feasible program of problem (7.1)-(7.3). By construction, however, this program is a support program.

The basis of the initial support program of problem (7.1)-(7.3) obtained by this technique may contain not only restraint vectors corresponding to the principal variables, but also unit vectors corresponding to the additional variables. Naturally, to the unit vectors of the basis correspond zero

components of the program. An initial support program whose basis contains unit vectors corresponding to the additional variables is a degenerate support program of the problem.

The rank of the matrix of coefficients of (7.2) is obviously m if the basis of the initial support program contains only restraint vectors corresponding to the principal variables of problem (7.1)–(7.3).

If additional unit vectors appear in the basis of the initial support program, the rank of the matrix $\|a_{ij}\|$ may be less than m. The rank, r, naturally cannot be less than the number of restraint vectors of problem (7.1)–(7.3) appearing in the basis of the initial support program. In Chapter 5, § 8 we shall show how to complete this set of linearly independent restraint vectors A_j to the maximum system of linearly independent vectors of the matrix

$$A=(A_1, A_2, \ldots, A_n)=\| a_{ij}\|_{mn}.$$

To the new restraint vectors introduced into the basis correspond zero components of the program. If the rank, r, of the matrix $\| a_{ij}\|$ is m, the basis obtained comprises m vectors. If the rank $r<m$, the basis of the initial support program also consists of less than r vectors. In this case, $m-r$ restraints linearly dependent on other equations of system (7.2) can be eliminated during the construction of the initial support program thus reducing problem (7.1)–(7.3) to a similar linear-programming problem in n variables and r equality restraints.

In case (b), when $\tilde{L}^*<0$, problem (7.1)–(7.3) has no feasible programs, i. e., the restraints of the initial problem are inconsistent. This can easily be demonstrated by reductio ad absurdum. Indeed, assume the problem to have at least one feasible program. Under this assumption, the auxiliary problem has a feasible program whose first n components coincide with the corresponding components of the initial program, and the remaining m components are all zero. The linear form vanishes on this program, which contradicts the assumption that the maximum value $\tilde{L}^*(X)$ is negative.

An initial support program of any linear-programming problem can thus be computed by solving the auxiliary problem with an obvious initial program. If the problem restraints happen to be inconsistent, this inconsistency is detected during the computations.

7-3. It is not always expedient to divide the solution of a linear-programming problem into two stages, i. e., computation of an initial support program and determination of an optimal program. Let us see how these two stages can be combined. Essentially, the solution of the initial linear-programming problem is replaced by the solution of an augmented problem with a larger set of support programs (one of which is always obvious) and having the same solutions, i. e., the same optimal programs, as the initial problem.

Thus, together with the initial linear-programming problem, consider the following augmented problem:

Maximize the linear form

$$L_M(\tilde{X})=\sum_{j=1}^{n} c_j x_j - M\sum_{i=1}^{m} x_{n+i},$$

where $M>0$ is a sufficiently large number, and

$$\tilde{X}=(x_1, x_2, \ldots, x_{n+m})$$

satisfies restraints (7.5)–(7.6).

We shall refer to the augmented problem as the M-p r o b l e m. We shall prove three theorems necessary to justify this new approach.

Theorem 7.1. If in the optimal program $\tilde{X}^*$ *of the* M*-problem* $x^*_{n+i}=0$ $(i=1,\ldots,m)$, *i.e., if* $\tilde{X}^*=(x^*_1,\ldots,x^*_n,0,\ldots,0)$ *the program* $X^*=(x^*_1, x^*_2,\ldots,x^*_n)$ *is a solution of the initial problem.*

Proof. If $\tilde{X}^*$ satisfies the restraints of the augmented problem, X^* satisfies the restraints of the initial problem. Hence, X^* is a feasible program of the initial problem.

Let us establish optimality. Assume that a feasible program $X=(x_1, x_2,\ldots,x_n)$ of problem (7.1)–(7.3) exists such that

$$L(X)>L(X^*);$$

$\tilde{X}=(x_1,\ldots,x_n,0,\ldots,0)$ is a feasible program of the M-problem. Hence,

$$L_M(\tilde{X})=L(X)>L(X^*)=L_M(\tilde{X}^*).$$

This equality contradicts the assumption that $\tilde{X}^*$ is a solution of the M-problem.

Thus, if in an optimal program of the M-problem $x^*_{n+i}=0$ for $i=1, 2,\ldots,m$, the first n components of this program specify a solution of the initial problem. This completes the proof.

Theorem 7.2. There always exists a number $M_1>0$, *such that for any* $M>M_1$ *the existence of at least one support program of the initial problem implies*

$$x^*_{n+i}=0 \qquad (i=1, 2,\ldots,m),$$

for any support solution of the M*-problem (if the latter is solvable).*

Proof. Let the proposition be false, i.e., assume that a support program $X=(x_1,\ldots,x_n)$ of the initial problem exists but that there is no M_1 such that for all $M>M_1$ $x^*_{n+i}=0$ $(i=1, 2,\ldots,m)$ for the support solution $\tilde{X}^*$ of the M-problem.

We introduce numbers $\underline{m}$ and $\bar{m}$ as follows. Let $\underline{m}$ be the minimum of the sum $\sum_{i=1}^{m} x_{n+i}$ on all the support programs of the M-problem for which this sum does not vanish. This minimum $\underline{m}>0$ exists, since the number of support programs of the M-problem is finite, and is independent of M. Let $\bar{m}$ be the maximum of the linear form $\sum_{j=1}^{n} c_j x_j$ in the set of the support programs of the M-problem. Obviously

$$\bar{m}\geqslant L(X).$$

Let

$$M_1=\frac{\bar{m}-L(X)}{\underline{m}}$$

and let us take $M>M_1$. By assumption, at least one of the components x^*_{n+i} $(i=1,\ldots,m)$ of the optimal program $\tilde{X}^*=(x^*_1,\ldots,x^*_{n+m})$ of the M-program is positive. Therefore, from the definition of $\underline{m}$ and $\bar{m}$,

$$\sum_{i=1}^{m} x^*_{n+i}\geqslant \underline{m}, \quad \sum_{j=1}^{n} c_j x^*_j \leqslant \bar{m}.$$

Hence,

$$L_M(X^*)=\sum_{j=1}^{n} c_j x^*_j - M\sum_{i=1}^{m} x^*_{n+i}\leqslant \bar{m}-M\underline{m}.$$

On the other hand, filling the program X of the initial problem with

zeros, we obtain a program $\tilde{X}=(x_1, \ldots, x_n, 0, \ldots, 0)$ of the M-problem. Here

$$L_M(\tilde{X})=L(X), \quad L_M(\tilde{X}) \leqslant L_M(X^*),$$

since X^* is a solution of the M-problem.

Applying the last three relationships we obtain

$$L(X) \geqslant \overline{m} - M\underline{m},$$

or (since $\underline{m} > 0$)

$$M \leqslant \frac{\overline{m} - L(X)}{\underline{m}} = M_1.$$

This inequality contradicts the condition $M > M_1$. This completes the proof.

Theorem 7.3. There exists a number M_2 such that for $M > M_2$ solvability of the initial problem (7.1)-(7.3) implies solvability of the associated M-problem.

The proof is based on certain features of the duality theory presented in Chapter 3. Consider two linear-programming problems, dual with respect to the initial problem and the M-problem, respectively:

1. Minimize the linear form

$$\sum_{i=1}^{m} b_i y_i \tag{7.7}$$

subject to the conditions

$$\sum_{i=1}^{m} a_{ij} y_i \geqslant c_j, \qquad j=1, 2, \ldots, n. \tag{7.8}$$

2. Minimize linear form (7.7) subject to conditions (7.8) and

$$y_i \geqslant -M, \qquad i=1, 2, \ldots, m. \tag{7.9}$$

Let $A_{j_1}, A_{j_2}, \ldots, A_{j_m}$ be a system of linearly independent restraint vectors. We determine the vector $Y_I=(y_1, y_2, \ldots, y_m)_I$ from the equations

$$\sum_{i=1}^{m} a_{ij_s} y_i = c_{j_s}, \qquad j_s \in I=(j_1, j_2, \ldots, j_m). \tag{7.10}$$

Let y_I equal the smallest components of the vector Y_I and let

$$M_2 = \max_I (-y_I),$$

where the maximum is taken over all the possible systems of m linearly independent restraint vectors.

Assume now that $M > M_2$ and that problem (7.1)-(7.3) is solvable. In this case, from the first duality theorem, the dual problem (7.7)-(7.8) is also solvable.

Let $Y=(y_1, y_2, \ldots, y_m)$ be a support program of this problem. Its components satisfy equations (7.10) for some set $I=(j_1, j_2, \ldots, j_m)$.

From the definition of M_2

$$y_i \geqslant -M_2 > -M, \qquad i=1, 2, \ldots, m.$$

Hence restraints (7.8), (7.9) are consistent and problem (7.7)-(7.9), differing from the solvable problem (7.7)-(7.8) in the additional restraints (7.9), is also solvable.

Applying the first duality theorem, we conclude that the M-problem (dual with respect to problem (7.7)-(7.9)) is solvable. This completes the proof.

The preceding results indicate a procedure for solving a linear-programming problem whose initial support program is not available to begin with. Rather than solve the initial problem (7.1)–(7.3), we apply the simplex method to the corresponding M-problem where

$$M \geqslant \max(M_1, M_2). \tag{7.11}$$

An initial support program of the M-problem is, obviously,

$$\tilde{X}_0 = (\underbrace{0, 0, \ldots, 0}_{n}, b_1, b_2, \ldots, b_m).$$

The basis constructed of unit vectors for the artificial variables x_{n+i}, $i=1, \ldots, m$ is generally called an artificial basis.

The simplex method starting from a support program with an artificial basis leads, after a finite number of iterations, either to case (a) (optimal program) or to case (b) (unsolvable problem).

Let now $\tilde{X}^* = (x_1^*, \ldots, x_{n+m}^*)$ be the last support program constructed during the solution of the M-problem. If case (a) applies, $\tilde{X}^*$ is a solution of the M-problem. Here two possibilities must be considered:

(i) $x_{n+1}^* = \ldots = x_{n+m}^* = 0$; then from Theorem 7.1 $X^* = (x_1^*, x_2^*, \ldots, x_n^*)$ is an optimal program of the initial problem (7.1)–(7.3);

(ii) $\sum_{i=n+1}^{n+m} x_i^* > 0$; from Theorem 7.2 and from condition (7.11) we see that problem (7.1)–(7.3) has no feasible programs, i.e., it is unsolvable.

If the process of solution of the M-problem terminates in case (b), then according to Theorem 7.3 and inequality (7.11) problem (7.1)–(7.3) is a priori unsolvable. Observe that in this case unsolvability of the initial problem may be due either to unboundedness of the linear form (7.1) in the set of feasible programs of the problem or to the fact that this set is empty. Construction of the corresponding examples is left to the reader (see Exercise 9).

Thus, the simplex method applied to the M-problem with M satisfying condition (7.11) gives, after a finite number of steps, a solution of the initial problem, or else proves its unsolvability. Here it is not required to know in advance an initial support program of the problem in question, (7.1)–(7.3). The above method of solving linear-programming problems is generally called the M-method.

We emphasize that to solve problems by the M-method there is no need to compute $M_0 = \max(M_1, M_2)$. When the evaluations Δ_j are determined while solving the M-problem, M should be taken greater than any other number used in comparison. If the solution of the M-problem yields an optimal program with $x_{n+i}=0$ for $i=1, 2, \ldots, m$, the first m components of the program specify a solution of the initial problem. If at least one of the components x_{n+i} of the optimal program is positive for any sufficiently large M, the initial program is unsolvable. Unsolvability of the M-problem implies unsolvability of the initial problem.

An example illustrating the application of the M-method will be given in Chapter 5, § 8. There we give also some computational procedures associated with the realization of the M-method.

The arguments presented in this section are with reference to linear-programming problems in canonical form. All the methods discussed here apply also to problems with bilateral restraints. In this case the auxiliary and the augmented problems should, obviously, be solved following the

rules given in § 5. Observe that additional (artificial) variables of the auxiliary (augmented) problems are only bounded below ($\alpha_j = 0$, $\beta_j = \infty$).

§ 8. Theoretical applications of the simplex method

The simplex method described in detail in the preceding sections of this chapter should not be considered as merely a computational procedure for solving concrete linear-programming problems. It is also most useful if we want to obtain various qualitative results of the linear-programming theory. In this section we shall illustrate the possibilities of the simplex method in application to the proof of several important theorems of linear programming, which have been proved in Chapters 2 and 3 on the basis of different considerations. All the results are stated and proved with reference to the canonical form of linear-programming problems (problem (1.1)–(1.3)). The rank of the matrix $A = (A_1, A_2, \ldots, A_n)$ is assumed to be m.

Theorem 8.1. (Existence of support program.) If the set of feasible programs of problem (1.1)–(1.3) is nonempty, it has at least one support program.

Proof. Without loss of generality we may take all the components of the constraint vector $B = (b_1, b_2, \ldots, b_m)^T$ to be nonnegative. We associate with problem (1.1)–(1.3) the following auxiliary linear-programming problem:

Maximize the linear form

$$L = -\sum_{i=1}^{m} x_{n+i} \tag{8.1}$$

subject to the conditions

$$\sum_{j=1}^{n} a_{ij}x_j + x_{n+i} = b_i, \qquad i = 1, 2, \ldots, m; \tag{8.2}$$

$$x_j \geqslant 0, \qquad j = 1, 2, \ldots, n+m. \tag{8.3}$$

Obviously, the $(m+n)$-dimensional vector

$$\bar{X}_0 = (0, 0, \ldots, 0, b_1, b_2, \ldots, b_m)$$

is a support program of problem (8.1)–(8.3). Therefore, when solving the auxiliary problem by the simplex method we may adopt it as the initial program. If in the degenerate case we apply the rule discussed in § 6 (here the vectors $R_1, R_2, \ldots, R_m$ can be taken as the unit vectors $e_1, e_2, \ldots, e_m$ comprising the basis of program $\bar{X}_0$), the solution of problem (8.1)–(8.3) is obtained by the simplex method in a finite number of iterations.

The linear form (8.1) is obviously limited by 0 in the set of feasible programs of the auxiliary problem. Hence, the solution cannot terminate in case (b) (see § 2) and will necessarily lead to a support solution of the problem (case (a)). Let this solution be

$$\bar{X}_1 = (x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}, x_{n+1}^{(1)}, \ldots, x_{n+m}^{(1)}).$$

By assumption, problem (1.1)–(1.3) has a certain program

$$X = (x_1, x_2, \ldots, x_n).$$

The vector

$$\bar{X} = (x_1, x_2, \ldots, x_n, 0, 0, \ldots, 0)$$

is therefore a program of problem (8.1)–(8.3). This program is, obviously, associated with the zero value of the linear form (8.1) and, consequently, is a solution of the auxiliary problem. Then $\bar{X}_1$ also makes (8.1) vanish, i.e.,

$$\sum_{i=n+1}^{n+m} x_i^{(1)} = 0.$$

Hence, since the components $x_i^{(1)}$ are nonnegative, we obtain

$$x_i^{(1)} = 0 \quad \text{for} \quad i = m+1, \ldots, m+n,$$

so that the vector

$$X_1 = (x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)})$$

is a program of problem (1.1)–(1.3).

By construction $\bar{X}_1$ is a support program of problem (8.1)–(8.3). Hence, the system of vectors A_j corresponding to $x_j^{(1)}>0$ $(j=1, 2, \ldots, n)$ is linearly independent. X_1 is therefore a support program of problem (1.1)–(1.3). This completes the proof.

Theorem 8.2. (Existence of support solution.) If a linear-programming problem (1.1)–(1.3) is solvable, there is a support solution among its solutions.

Proof. Since the set of feasible programs of problem (1.1)–(1.3) is nonempty, the problem, according to the Theorem 8.1, has a support program

$$X=(x_1, x_2, \ldots, x_n).$$

Taking X as the initial program, we apply the simplex procedure.

Since the problem is solvable, a finite number of iterations will produce the support solution. This completes the proof.

Theorem 8.3. (Solvability of the linear-programming problem.) The linear-programming problem is solvable if and only if

(i) the set of its feasible programs is nonempty;

(ii) the linear form of the problem is bounded in this set.

Proof. Necessity is obvious. We shall prove sufficiency.

From (i) and Theorem 8.1 the problem has a support program. As in the proof of Theorem 8.2, we take this program as the initial program and apply the simplex procedure. The solution should obviously terminate in case (a) (case (b) is ruled out by condition (ii)). Hence, the program constructed in the one but last iteration is a solution of the problem. This completes the proof.

In conclusion we offer a simple proof of the first duality theorem. First let us recall the statement of the problem dual with respect to problem (1.1)–(1.3):

Minimize the linear form

$$\sum_{i=1}^{m} b_i y_i \tag{8.4}$$

subject to the conditions

$$\sum_{i=1}^{m} a_{ij} y_i \geqslant c_j, \qquad j=1, 2, \ldots, n. \tag{8.5}$$

First duality theorem. Solvability of problem (1.1)–(1.3) implies solvability of the dual problem (8.4)–(8.5). Here the maximum of linear form (1.1) subject to conditions (1.2), (1.3) coincides with the minimum of linear form (8.4) subject to conditions (8.5).

Proof. Taking some support program of problem (1.1)–(1.3) as the initial program we apply the simplex procedure.

Since the problem is solvable, the solution process, in a finite number of iterations, yields the support optimal program

$$X=(x_1, x_2, \ldots, x_n),$$

which satisfies the requirements of the optimality test (case (a)). This indicates that the vector $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$ determined from the system of equations

$$\sum_{j=1}^{m} a_{ij}\lambda_i=c_j, \quad j\in I_X, \tag{8.6}$$

has the property

$$\sum_{i=1}^{m} a_{ij}\lambda_i \geqslant c_j, \quad j\notin I_X \tag{8.7}$$

(here the second form of the test is used; the vectors A_j, $j\in I_X$ constitute the basis of program X). Relationships (8.6) and (8.7) show that the vector Λ satisfies restraints (8.5), i.e., is a program of problem (8.4)–(8.5).

We shall show that

$$\sum_{j=1}^{n} c_j x_j=\sum_{i=1}^{m} b_i\lambda_i. \tag{8.8}$$

Indeed

$$\sum_{j=1}^{n} c_j x_j=\sum_{j\in I_X} c_j x_j=\sum_{j\in I_X} x_j \sum_{i=1}^{m} a_{ij}\lambda_i=\sum_{j=1}^{n} x_j \sum_{i=1}^{m} a_{ij}\lambda_i=\sum_{i=1}^{m} \lambda_i \sum_{j=1}^{n} a_{ij}x_j=\sum_{i=1}^{m} \lambda_i b_i.$$

The first and the third equalities hold since $x_j=0$ for $j \notin I_{X^*}$. The second and the fifth equalities follow, respectively, from (8.6) and (1.2). The fourth equality is simply the result of changing the order of summation.

Now let $Y=(y_1, y_2, \ldots, y_m)$ be a feasible program of problem (8.4)–(8.5). Then

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{j=1}^{n} x_j \sum_{i=1}^{m} a_{ij} y_i = \sum_{i=1}^{m} y_i \sum_{j=1}^{n} a_{ij} x_j = \sum_{i=1}^{m} y_i b_i.$$

The first inequality here is a consequence of restraints (8.5), which vector Y satisfies. The other relationships hold for the same reasons as the fourth and the fifth in the previous chain of equalities. Thus,

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{i=1}^{m} y_i b_i, \tag{8.9}$$

where Y is a feasible program of problem (8.4)–(8.5).

Relationships (8.8) and (8.9) show that Λ is an optimal program of problem (8.4)–(8.5). The dual problem of problem (1.1)–(1.3) is solvable, and the optimal programs of the two problems are related by equality (8.8). This completes the proof.

The preceding examples of proofs of several important theorems of linear programming show that the simplex procedure is a highly effective tool in qualitative investigations in the linear-programming theory. It must be emphasized that the application of this procedure always leads to constructive proof which elicits not only the fact, but also the feasible computational approaches.

EXERCISES TO CHAPTER 4

1. Prove the necessity of the optimality test (in the first or the second form) for the case of nondegenerate support program.
2. Give an example of a problem in which the requirements of the optimality test are not necessary.
3. Show that the linear form of problem (1.1)–(1.3) with a nonempty set of feasible programs is unbounded if and only if the positive semiaxis Ou_{m+1} belongs to the cone spanned by the augmented restraint vectors.
4. Show that the concept of a support program introduced in § 5 for a problem with bilateral restraints corresponds to the concept of a support program for a linear-programming problem in arbitrary form, given in Chapter 2, § 4.
5. Prove that the concept of nondegeneracy introduced in § 5 for a problem with bilateral restraints corresponds to the general concept of nondegeneracy given in Chapter 2, § 4.
6. Prove that

(a) any system of m linearly independent restraint vectors of problem (5.1)–(5.3) may serve as the basis of at most 2^{n-m} support programs of this problem;

(b) a support program of problem (5.1)–(5.3) whose degree of degeneracy is μ has at most $C^{\mu}_{n-m+\mu}$ different bases.

Give examples in which the maximum estimates are attained. (The degree of degeneracy of a support program is $\mu=m-\nu$, where ν is the number of basis components different from the bounds values of the corresponding variables.)

7. Prove that two problems which have feasible programs and differ only in the constraint vectors are simultaneously solvable or unsolvable.
8. Prove that a cycle is impossible in a problem whose support programs all have degree of degeneracy not exceeding 1.
9. Show by examples that when the linear-programming problem is solved by the M-method, case (b) may be due either to unboundedness of the linear form or to inconsistency of the problem restraints.
10. Prove with the aid of the simplex procedure that nondegeneracy of the linear-programming problem implies unique solvability of the dual problem.

Chapter 5

THE SIMPLEX COMPUTATIONAL PROCEDURE

In this chapter we describe the computational procedures for solving linear-programming problems by the simplex method. This will be done in the following order. In § 1 relationships between the parameters of support programs related by elementary transformations are established. Recurrence formulas for the parameters of successive iterations are derived. In §§ 2-6 the computational procedures associated with the two algorithms of the simplex method are given in detail. All computational procedures are illustrated by numerical examples.

The simplex method has been perfected to a higher degree than other procedures and the literature deals fairly extensively with the subject. In different sources we find different approaches to the method and analysis of the corresponding algorithms. In § 4, which deals with the vector and the coordinate form of the method, the equivalence of the different approaches to the simplex method is established, and the meaning of the transformations in simplex algorithms is elucidated.

In § 7 the specific features of solution of linear-programming problems with bilaterally restrained variables are dealt with. In § 8 the sequence of computations leading to the determination of an initial support program is described. The last section, § 9, deals with a relatively special question. The possibility of cycling in degenerate problems is studied and an example illustrating this phenomenon is constructed.

§ 1. Relationship between the parameters of successive iterations

1-1. The simplex method involves a sequence of successive elementary transformations of a given support program of the problem to another support program which is closer to the optimum. In order to test the support program for optimality and improve it in each successive step, we compute, in each iteration, various parameters which help characterize and evaluate the program.

In the preceding chapter we obtained systems of equations relating the simplex parameters with the problem restraints (the restraint vectors, the constraint vector, and the linear-form coefficients). As has been indicated, the solution of these systems of equations constitutes the most tedious stage of computations. However, the specific properties of elementary transformations make it possible to confine the solution of systems of equations to the first stage, in which the initial data are prepared for the first

iteration. Applying the techniques of Chapter 4, § 7, we distinguish between the determination of the initial support program and all its characterizing parameters, on the one hand, and the solution of systems of equations, on the other. It is far more tedious to determine the parameters of the initial support program than to compute the parameters of all the subsequent iterations, since the latter are related by very simple recurrence formulas.

The relationships between the simplex parameters for different algorithms are established according to the same principle. Below we prove a fairly general theorem from which we derive, as corollaries, all the recurrence formulas needed for solving the problem according to the various simplex computational procedures.

Consider two systems of linearly independent vectors $\{P_i\}_{i\in I}$ and $\{P_i\}_{i\in I'}$. The sets of indices I and I' differ in only one element. The set I' is obtained from the set I by substituting the index k for the index r, so that

$$\left\{ {I \atop i\neq r} \right\} \equiv \left\{ {I' \atop i\neq k} \right\}. \tag{1.1}$$

Let ζ_{ij} and ζ'_{ij} be the coefficients appearing when an arbitrary system of vectors Q_j is expressed in terms of the two systems of linearly independent vectors $\{P_i\}_{i\in I}$ and $\{P_i\}_{i\in I'}$, i.e.,

$$Q_j = \sum_{i\in I} \zeta_{ij} P_i, \tag{1.2}$$

$$Q_j = \sum_{i\in I'} \zeta'_{ij} P_i. \tag{1.3}$$

Let the coefficients η_{ik} appearing when the vector P_k is expressed in terms of the system $\{P_i\}_{i\in I}$ be known:

$$P_k = \sum_{i\in I} \eta_{ik} P_i. \tag{1.4}$$

Here $\eta_{rk} \neq 0$, since the two systems $\{P_i\}_{i\in I}$ and $\{P_i\}_{i\in I'}$ comprise linearly independent vectors and differ only in the vector $P_r (P_k)$.

We shall now establish a relationship between ζ'_{ij}, on the one hand, and ζ_{ij} and η_{ik} on the other.

We have the following proposition.

Theorem 1.1. *The coefficients ζ_{ij} and ζ'_{ij} in the expansion of arbitrary vectors Q_j in terms of the systems $\{P_i\}_{i\in I}$ and $\{P_i\}_{i\in I'}$, which satisfy relationships (1.1) and (1.4), are related by the formulas*

$$\zeta'_{ij} = \begin{cases} \zeta_{ij} - \dfrac{\zeta_{rj}}{\eta_{rk}} \eta_{ik} & \text{for } i\neq k, \\ \dfrac{\zeta_{rj}}{\eta_{rk}} & \text{for } i = k. \end{cases} \tag{1.5}$$

Proof. From equality (1.4) we have

$$P_k = \sum_{\substack{i\in I \\ i\neq r}} \eta_{ik} P_i + \eta_{rk} P_r.$$

By assumption $\eta_{rk}\neq 0$. Therefore, applying (1.1) and (1.2), we obtain

$$Q_j = \sum_{\substack{i\in I \\ i\neq r}} \zeta_{ij} P_i + \frac{\zeta_{rj}}{\eta_{rk}} \left(P_k - \sum_{\substack{i\in I \\ i\neq r}} \eta_{ik} P_i \right) = \sum_{\substack{i\in I' \\ i\neq k}} \left(\zeta_{ik} - \frac{\zeta_{rj}}{\eta_{rk}} \eta_{ik} \right) P_i + \frac{\zeta_{rj}}{\eta_{rk}} P_k.$$

The system $\{P_i\}_{i\in I'}$ is linearly independent. Therefore, the coefficients of

the vector Q_j expressed in terms of the vectors of this system are uniquely defined. Comparing the last relationship with (1.3), we obtain (1.5). This completes the proof.

The vector P_k in the system $\{P_i\}_{i \in I'}$ occupies the same position as the vector P_r in the system $\{P_i\}_{i \in I}$. We have already seen that in some cases the indices of the components are most suitably related with the positions of the corresponding vectors in the system, rather than with the actual indices of these vectors. In view of this we rewrite the recurrence formulas (1.5) in the form

$$\zeta'_{lj} = \begin{cases} \zeta_{lj} - \frac{\zeta_{rj}}{\eta_{rk}} \eta_{lk} & \text{for } l \neq r, \\ \frac{\zeta_{rj}}{\eta_{rk}} & \text{for } l = r. \end{cases} \tag{1.6}$$

Here we assume that P_r occupies the r-th position in the system $\{P_i\}_{i \in I}$.

1-2. We shall use (1.6) to derive recurrence formulas for the parameters of successive iterations in the first simplex algorithm.

For the first algorithm, associated with the first form of the optimality test of support programs, in each iteration the following parameters must be computed:

(a) x_{ij}, the coefficients in the expansion of any restraint vector A_j ($j=1, 2, \ldots, n$) in terms of the basis vectors;

(b) x_{i0}, the basis components of the support program ($i=1, 2, \ldots, m$);

(c) Δ_j, the evaluations of the restraint vectors A_j with respect to the basis in question ($j=1, 2, \ldots, n$);

(d) $L(X)$, the value of the linear form on program X.

Let

$$V = -\bar{e}_{m+1},$$

where $\bar{e}_{m+1} = (0, \ldots, 0, 1)$ is the $(m+1)$-dimensional unit vector. Let

$$\{P_i\}_{i \in I} = \{(\bar{A}_i)_{i \in I_X}, V\}, \tag{1.7}$$

$$\{P_i\}_{i \in I'} = \{(\bar{A}_i)_{i \in I_{X'}}, V\}. \tag{1.8}$$

The first system consists of the m augmented restraint vectors, specified by the basis of the support program X, and the $(m+1)$-dimensional vector V. The second system consists of the augmented vectors, corresponding to the support program X', and the vector V. The support program X' is obtained from the support program X by the elementary transformation associated with the vector A_k. The two systems differ in one vector. The vector $\bar{A}_k$ in the second system occupies the r-th position in the basis. In the first system the vector $\bar{A}_{s_r}$ is in this position. The vectors of both the first and the second systems are, obviously, linearly independent.

We take as the vector Q_j any augmented restraint vector $\bar{A}_j$, $j=0, 1, 2, \ldots, n$. (The augmented vector $\bar{A}_0$ is defined as the $(m+1)$-dimensional vector whose first m components coincide with the corresponding components of the constraint vector, and whose $(m+1)$-th component c_0 is zero.)

The expansion of the vector $\bar{A}_j$ in terms of the vectors of the first system can be written as follows:

$$\bar{A}_j = \sum_{l=1}^{m} \zeta_{lj} \bar{A}_{s_l} + \zeta_{m+1,j} V, \quad j=0, 1, \ldots, n. \tag{1.9}$$

Analogously, the expansion of the vector $\bar{A}_k$ (the equivalent of the vector P_k

in Theorem 1.1) is:

$$\bar{A}_k = \sum_{i=1}^{m} \eta_{ik}\bar{A}_{s_i} + \eta_{m+1,k}V. \tag{1.10}$$

The first m components of the augmented vectors specify the restraint vectors and are related by

$$A_j = \sum_{i=1}^{m} x_{ij}A_{s_i}, \qquad j=0, 1, 2, \dots, n,$$

and, correspondingly,

$$A_k = \sum_{i=1}^{m} x_{ik}A_{s_i}.$$

Since the first m components of the vector V are all zero, we have

$$\zeta_{ij} = x_{ij}, \qquad i=1, 2, \dots, m, \quad j=0, 1, \dots, n,$$
$$\eta_{ik} = x_{ik}, \qquad i=1, 2, \dots, m.$$

Analogously,

$$\zeta'_{ij} = x'_{ij}, \quad i=1, 2, \dots, m, \quad j=0, 1, \dots, n.$$

To compute $\zeta_{m+1,j}$ and $\eta_{m+1,k}$ for $j=0, 1, \dots, n$ we project both sides of equalities (1.9) and (1.10) onto the $\bar{e}_{m+1}$-axis or, equivalently, compute the $(m+1)$-th components of the vectors in the left- and the right-hand sides of each of these equalities:

$$c_j = \sum_{i=1}^{m} x_{ij}c_{s_i} - \zeta_{m+1,j},$$
$$c_k = \sum_{i=1}^{m} x_{ik}c_{s_i} - \eta_{m+1,k}.$$

Comparing these formulas with (2.1) of the previous chapter and since $c_0 = 0$, we obtain

$$\zeta_{m+1,j} = \Delta_j, \quad j=1, 2, \dots, n, \quad \eta_{m+1,k} = \Delta_k;$$
$$\zeta_{m+1,0} = \Delta_0 = L(X).$$

Similarly,

$$\zeta'_{m+1,j} = \Delta'_j, \quad j=1, 2, \dots, n;$$
$$\zeta'_{m+1,0} = \Delta'_0 = L(X').$$

The following notations are obvious:

$$V = \bar{A}_{s_{m+1}}; \quad \Delta_j = x_{m+1,j}, \quad j=1, 2, \dots, n; \quad \Delta_0 = L(X) = x_{m+1,0};$$

and, correspondingly,

$$\Delta'_j = x'_{m+1,j}, \quad j=1, 2, \dots, n; \quad \Delta'_0 = L(X') = x'_{m+1,0}.$$

In these notations, (1.9) and (1.10) become

$$\left.\begin{aligned} \bar{A}_j &= \sum_{i=1}^{m+1} x_{ij}\bar{A}_{s_i}, \quad j=0, 1, \dots, n, \\ \bar{A}_k &= \sum_{i=1}^{m+1} x_{ik}\bar{A}_{s_i}. \end{aligned}\right\} \tag{1.11}$$

From the definition of the elementary transformation we have $\eta_{rk} = x_{rk} \neq 0$ (the index r gives the position of the basis on which the ratio $\frac{x_{i0}}{x_{ik}}$, with $x_{ik} > 0$, reaches its minimum).

All the requirements of Theorem 1.1 are thus satisfied. Applying the propositions of this theorem to the expansion coefficients of the augmented

restraint vectors in terms of the vectors of systems (1.7) and (1.8), we obtain the following result:

The parameters of two successive iterations are related by the recurrence formulas

$$x'_{ij}=\begin{cases} x_{ij}-\frac{x_{rj}}{x_{rk}}x_{ik} & \text{for } i\neq r, \\ \frac{x_{rj}}{x_{rk}} & \text{for } i=r, \\ i=1,\ 2,\ \ldots,\ m+1, & j=0,\ 1,\ 2,\ \ldots,\ n. \end{cases} \tag{1.12}$$

Formulas (1.12) are quite sufficient for proceeding from a given step of the simplex method to the successive one. Recurrence formulas (1.12) make it possible to compute the following parameters of each iteration (provided the parameters of the preceding approximation are available):

(a) the expansion coefficients of all the restraint vectors expressed in terms of the basis vectors ($i=1,\ 2,\ \ldots,\ m;\ j=1,\ 2,\ \ldots,\ n$);

(b) the basis components of the support program ($i=1,2,\ldots,m;\ j=0$);

(c) the relative evaluations of the restraint vectors ($i=m+1;\ j=1,\ 2,\ \ldots,\ n$);

(d) the value of the linear form ($i=m+1;\ j=0$).

Formulas (1.12) constitute the basis of the first simplex algorithm.

1-3. We now use (1.6) to derive recurrence formulas for the parameters of the successive iterations in the second simplex algorithm.

The second algorithm is based on the second optimality test for support programs in which the evaluations of the restraint vector, Δ_j, are computed in each iteration with the aid of the vector $\Lambda=(\lambda_1,\ldots,\lambda_m)$. The parameters λ_i, in turn, satisfy the equations

$$\sum_{i=1}^{m} a_{ij}\lambda_i=c_j,\quad j\in I_X. \tag{1.13}$$

Computation of λ_i involves finding the inverse of the matrix A_X.

To solve problems by the second algorithm, we must compute the following parameters in each iteration:

(a) e_{ij}, the elements of the inverse matrix A_X^{-1};

(b) λ_i, evaluations of problem restraints with respect to the basis of the support program X;

(c) x_{i0}, the basis components of the support program;

(d) $L(X)$, the value of the linear form on program X.

Observe that it is convenient to take the parameters e_{ij} – the elements of the matrix A_X^{-1} – as the coefficients of the m-dimensional unit vectors

$$e_j=\Bigl(\overbrace{\underbrace{0,\ \ldots,\ 0,\ 1,}_{j}\ 0,\ \ldots,\ 0}^{m}\Bigr)$$

expanded in terms of basis vectors.

Let, as before, systems $\{P_i\}_{i\in I}$ and $\{P_i\}_{i\in I'}$ be defined by (1.7) and (1.8). The vectors Q_j here are taken as the $(m+1)$-dimensional unit vectors

$$\bar{e}_j=\Bigl(\overbrace{\underbrace{0,\ \ldots,\ 0,\ 1,}_{j}\ 0,\ \ldots,\ 0}^{m+1}\Bigr),\qquad j=1,\ 2,\ \ldots,\ m,$$

and

$$\bar{e}_0=\bar{A}_0=(b_1,\ \ldots,\ b_m,\ 0).$$

The expansion of the vector $\bar{e}_j$ in terms of the vectors of the first system has the form

$$\bar{e}_j=\sum_{i=1}^{m}\zeta_{ij}\bar{A}_{s_i}+\zeta_{m+1,\,j}V; \qquad j=0,\ 1,\ 2,\ \ldots,\ m. \tag{1.14}$$

The expansion of the m-dimensional vectors e_j in terms of the basis vectors (A_{s_i}) by definition has the form

$$e_j=\sum_{i=1}^{m}e_{ij}A_{s_i}, \qquad j=0,\ 1,\ 2,\ \ldots,\ m.$$

Hence

$$\zeta_{ij}=e_{ij}, \qquad i=1,\ 2,\ \ldots,\ m, \quad j=0,\ 1,\ \ldots,\ m$$

($e_{i_0}=x_{i_0}$ are the basis components of the support program). Similarly,

$$\zeta'_{ij}=e'_{ij}, \qquad i=1,\ 2,\ \ldots,\ m; \quad j=0,\ 1,\ \ldots,\ m.$$

To compute $e_{m+1,\,j}$ $(j=1,\ 2,\ \ldots,\ n)$, we project both sides of equality (1.13) onto the vector $\bar{e}_{m+1}$:

$$0=\sum_{i=1}^{m}e_{ij}c_{s_i}-\zeta_{m+1,\,j}.$$

Multiplying both sides of the last equality by a_{js_t} and summing over j from 1 to m, and since a_{ij} and e_{ij} $(i=1,\ \ldots,\ m;\ j=1,\ \ldots,\ m)$ are the elements of mutually inverse matrices, we have

$$0=c_{s_t}-\sum_{j=1}^{m}a_{js_t}\zeta_{m+1,\,j}, \qquad t=1,\ 2,\ \ldots,\ m.$$

Replacing the index j by i, and since the indices $s_t\,(t=1,\ 2,\ \ldots,\ m)$ constitute the set I_X, we have

$$\sum_{i=1}^{m}a_{ij}\zeta_{m+1,\,i}=c_j, \quad j\in I_X.$$

The vector $\{\zeta_{m+1,\,i}\}_i$ is, thus, the solution of system (1.13). Hence

$$\zeta_{m+1,\,j}=\lambda_j, \qquad j=1,\ 2,\ \ldots,\ m. \tag{1.15}$$

Taking $j=0$ in (1.14) and rewriting this equality for the $(m+1)$-th components of the vectors $\bar{A}_{s_i}$ and V, we have

$$0=\sum_{i=1}^{m}\zeta_{i0}c_{s_i}-\zeta_{m+1,\,0}.$$

But $\zeta_{i0}=e_{i_0}=x_{i_0}$, so that

$$\zeta_{m+1,\,0}=\sum_{i=1}^{m}x_{i_0}c_{s_i}=L(X)=\lambda_0.$$

The following notations are obvious:

$$\begin{gathered}V=\bar{A}_{s_{m+1}};\quad \lambda_j=e_{m+1,\,j},\quad j=1,\ \ldots,\ m;\\ \lambda_0=L(X)=e_{m+1,\,0},\end{gathered} \tag{1.16}$$

and, correspondingly,

$$\lambda'_j=e'_{m+1,\,j}, \quad j=1,\ \ldots,\ m; \quad \lambda'_0=L(X')=e'_{m+1,\,0}.$$

We have, moreover, assumed that

$$x_{i_0}=e_{i_0}, \quad i=1,\ 2,\ \ldots,\ m, \tag{1.17}$$

and, correspondingly,

$$x'_{i0}=e'_{i0}.$$

In the new notations, formula (1.14) becomes

$$\bar{e}_j=\sum_{i=1}^{m+1} e_{ij}\bar{A}_{s_i}, \qquad j=0,\ 1,\ 2,\ \ldots,\ m. \tag{1.18}$$

The expansion of the vectors $\bar{e}_j=Q_j$ in terms of the vectors of system (1.8) can be given by an analogous formula:

$$\bar{e}_j=\sum_{i=1}^{m+1} e'_{ij}\bar{A}_{s'_i}, \qquad j=0,\ 1,\ 2,\ \ldots,\ m. \tag{1.19}$$

Let $I=(s_1,\ s_2,\ \ldots,\ s_r,\ \ldots,\ s_m)$; $I'=(s'_1,\ s'_2,\ \ldots,\ s'_r,\ \ldots,s'_m)$;

$$s'_i=\begin{cases} s_i & \text{for} \quad i\neq r, \\ k & \text{for} \quad i=r. \end{cases}$$

The expansion of the vector $\bar{A}_k=P_k$ in terms of the vectors of the first system is written, from (1.11), in the form

$$\bar{A}_k=\sum_{i=1}^{m+1} x_{ik}\bar{A}_{s_i}. \tag{1.20}$$

Now, applying Theorem 1.1 to the expansion coefficients of the vectors $\bar{e}_j$ in the vectors of the systems (1.7) and (1.8) and applying formulas (1.18)–(1.20), we obtain the following recurrence formulas for the parameters of two successive iterations:

$$e'_{ij}=\begin{cases} e_{ij}-\dfrac{e_{rj}}{x_{rk}}x_{ik} & \text{for} \quad i\neq r, \\ \dfrac{e_{rj}}{x_{rk}} & \text{for} \quad i=r, \end{cases} \tag{1.21}$$

$$i=1,\ 2,\ \ldots,\ m+1, \quad j=0,\ 1,\ 2,\ \ldots,\ m.$$

Formulas (1.21) make it possible to compute the following parameters of each iteration, if the analogous parameters of the preceding approximation are known:

(a) the elements e_{ij} of the inverse matrix A_X^{-1} of the basis vectors ($i=1$, $2,\ \ldots,\ m$; $j=1,\ 2,\ \ldots,\ m$);

(b) the basis components of the support program $x_{i0}=e_{i0}$ ($i=1,\ \ldots,\ m$; $j=0$);

(c) the relative evaluations of the problem restraints $\lambda_j=e_{m+1,\,j}$ ($i=m+1$; $j=1,\ 2,\ \ldots,\ m$);

(d) the value of the linear form $L(X)=e_{m+1,\,0}$ ($i=m+1$; $j=0$).

The recurrence formulas (1.21) constitute the basis of the second simplex algorithm.

§ 2. The first simplex algorithm

2-1. The theoretical principles of the first algorithm were discussed in Chapter 4, § 2. Here we describe the sequence of computations to be followed when this algorithm is applied to the solution of linear-programming problems in canonical form.

We have already seen that the simplex method, proceeding from some initial support program and gradually improving it, produces, after a finite number of iterations, an optimal program or establishes unsolvability

of the problem. Each iteration involves corresponding transition from one tableau of the algorithm to the next.

Let us introduce some terms which will simplify the presentation. The concept row-row product or row-column product will refer to multiplication of the corresponding rows or a row and a column in the tableau by a scalar. We shall say that a row (column) is divided or multiplied by a number, if each element of the row (column) is divided or multiplied by this number. When saying that two rows (or two columns) are added or that one row is subtracted from the other, we shall imply that the operations specified are carried out on the corresponding elements of the rows or the columns.

Each iteration in the simplex procedure consists of two stages. In the first stage, the support program in question is tested for optimality. The second stage is carried out if the program considered is not optimal and the problem has not been shown to be unsolvable. In the second stage the elementary transformation, which produces a new support program with a higher value of the linear form, and the vector to be introduced into and that to be eliminated from the basis are determined. Then the basis components of the new support program and all the parameters needed for proceeding with the solution are computed.

Any vector whose evaluation with respect to the preceding basis is negative can be introduced into the new basis. The increase of linear form will be maximum in each iteration if we introduce into the basis the vector A_j for which $\theta_0^{(j)}\Delta_j$ is minimum. Experience shows that this, as a rule, also reduces the number of iterations needed to yield a solution. However, establishing for which vector the product $\theta_0^{(j)}\Delta_j$ is minimum is a fairly complicated procedure. First it is necessary to calculate

$$\theta_0^{(j)} = \min_{x_{ij} > 0} \frac{x_{i0}}{x_{ij}}$$

for all j such that $\Delta_j < 0$. Then the products $\theta_0^{(j)}\Delta_j$ must be compared. The index j corresponding to the minimum product specifies the vector to be introduced into the basis. Numerous computations show that generally the problem is solved faster if the simpler procedure is adopted, namely introduction into the basis of vector A_k with the least evaluation Δ_k. Here θ_0 is determined in each iteration only once and the number of computations in each iteration is considerably reduced. In the present section the computational procedure is described with reference to this simpler way of choosing the vector to be introduced into the basis.

The vector occupying the position of the basis on which

$$\theta_0 = \min_{x_{ik} > 0} \frac{x_{i0}}{x_{ik}}$$

is obtained (k is the index of the vector to be introduced into the basis) is eliminated from the basis. If θ_0 is obtained for several vectors, any of these is eliminated; to be specific, we may decide to eliminate, e.g., the vector with the least position index. This simple rule does not eliminate the possibility of cycling. However, cycling in linear-programming problems is an extremely rare phenomenon and the more cumbersome rule, which allows for the possibility of cycling (see 2-3), should be applied only once a cycle has been detected. Once the cycle has been eliminated, it is advisable to return to the simplified rule of determining the vector to be eliminated from the basis.

Having determined the index of the vector to be introduced into the basis and the position in the basis in which it is to be introduced (the position of the vector to be eliminated from the basis), we proceed with the computation of the parameters of the current support program. Recurrence formulas (1.12) are applied to determine the basis variables of the new support program, the expansion coefficients of the restraint vectors expressed in terms of the new basis vectors, the evaluations of the restraint vectors with respect to the new basis, and the value of the linear form on the new program. These computations prepare the ground for the next iteration.

2-2. We now describe the sequence of computations in a single iteration. Assume that the l-th iteration has been completed. The l-th tableau (see Table 5.1) is then filled, except for the last column.

TABLE 5.1

l-th tableau

№	C_X	B_X	A_0	A_1	A_2	...	A_k	...	A_n	θ
1	c_{s_1}	A_{s_1}	$x_{10}^{(l)}$	$x_{11}^{(l)}$	$x_{12}^{(l)}$	...	$x_{1k}^{(l)}$	...	$x_{1n}^{(l)}$	
2	c_{s_2}	A_{s_2}	$x_{20}^{(l)}$	$x_{21}^{(l)}$	$x_{22}^{(l)}$	...	$x_{2k}^{(l)}$	...	$x_{2n}^{(l)}$	
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	...	⋮	⋮
r	c_{s_r}	A_{s_r}	$x_{r0}^{(l)}$	$x_{r1}^{(l)}$	$x_{r2}^{(l)}$	...	$x_{rk}^{(l)}$	...	$x_{rn}^{(l)}$	θ_0
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	...	⋮	⋮
m	c_{s_m}	A_{s_m}	$x_{m0}^{(l)}$	$x_{m1}^{(l)}$	$x_{m2}^{(l)}$	...	$x_{mk}^{(l)}$	...	$x_{mn}^{(l)}$	
$m+1$	—	—	$L^{(l)}$	$\Delta_1^{(l)}$	$\Delta_2^{(l)}$	...	$\Delta_k^{(l)}$	...	$\Delta_n^{(l)}$	

The tableaus are filled as follows. When there are no entries, a dash is written and if the entries are zeroes, the cells are left empty.

In the first column (No.) of the tableau we give the row number. The first m rows are numbered so as to coincide with the basis positions. In the second column (C_X) the coefficients c_{s_i} of the linear form associated with the basis variables, are written.

The basis vectors A_{s_i} are entered into the column (B_X). In the next column $(A_0 = B)$ we write the basis components x_{i0} of the support program (the coefficients of the expansion of constraint vector B written in terms of the basis vectors). The columns $A_1, A_2, \ldots, A_n$ contain the coefficients x_{ij} of the corresponding expansion of the restraint vectors A_j expressed in terms of the basis vectors. Column elements refer to the entries in the first m rows of the tableau only. In the last, $(m+1)$-th, row we write the evaluations $\Delta_j^{(l)}$ of the restraint vectors A_j with respect to the basis. In the $(m+1)$-th cell of the column $A_0 = B$ we write the value of the linear form $L(X^{(l)})$. We remind

the reader that in the previous section we used the notations

$$\Delta_j^{(l)} = x_{m+1,\,j}^{(l)}; \quad L(X^{(l)}) = x_{m+1,\,0}.$$

The index l indicating the number of the iteration will be omitted wherever no confusion can arise.

The columns $A_0, A_1, \ldots, A_n$ (all the $m+1$ entries) constitute the principal part of the tableau. The principal part of the tableau is, thus, a matrix $\| x_{ij}^{(l)} \|$, where $i = 1, 2, \ldots, m+1$; $j = 0, 1, 2, \ldots, n$.

Computations made in the $(l+1)$-th iteration are written in the column θ of the -th tableau and form the principal part of the $(l+1)$-th tableau.

The first stage of the $(l+1)$-th iteration starts with examination of the $(m+1)$-th row of the l-th tableau. To establish whether case (a) applies (optimal program), we must isolate the vectors with negative evaluations. If all $\Delta_j = x_{m+1,\,j} \geqslant 0$ $(j = 1, 2, \ldots, n)$, the support program produced by the l-th iteration is an optimal program of the problem. Once this fact has been established, the solution process is terminated. Now let us assume that there are negative evaluations in the last row of the l-th tableau. To establish whether case (b) applies (unsolvable problem), we examine the columns A_j with $\Delta_j < 0$ and check the signs of the entries in these columns. If there exists at least one column A_j with $\Delta_j < 0$ for all $x_{ij} \leqslant 0$ $(i = 1, 2, \ldots, m)$, the problem is unsolvable (case (b)). Once this fact has been established, again the process terminates.

Case (c) applies if each column A_j with $\Delta_j < 0$ contains at least one positive x_{ij}. If so, we proceed with the second stage of the iteration.

The vector A_k with the last evaluation

$$\Delta_k = \min_j \Delta_j$$

is introduced into the basis. After A_k has been chosen, the last column of the l-th tableau, θ, is filled. In column θ we write the ratio of the basis variables x_{i0} (the entries in the first column $A_0 = B$) to the corresponding components x_{ik} (the entries in the column A_k). Only those cells of the column θ are filled which correspond to positive x_{ik}. The least element of the column θ is denoted by θ_0. The basis vector A_{s_r} on which θ_0 is obtained should be eliminated from the basis. If the support program $X^{(l+1)}$ is degenerate and, consequently, θ_0 is obtained on several basis positions, any basis vector in these positions is eliminated. If cycling occurs, apply the rule given in 2-3.

The column A_k, corresponding to the vector introduced into the basis, and the r-th column, corresponding to the vector eliminated from the basis, are marked, e.g., by a frame. The column A_k and the r-th row are called the direction column and the direction row of the transformation, respectively. The element x_{rk} in the cell which is the intersection of the direction column and the direction row is called the direction element.*

Having found the direction element, we proceed to fill the $(l+1)$-th tableau. In the r-th cell of column B_X we write the vector A_k which is now, according to its new position, denoted by A_{s_r} in the $(l+1)$-th tableau. In the other cells of the column B_X we write the same vectors as in the l-th tableau. The rows corresponding to the vector introduced into the basis in the preceding iteration and to the vector to be eliminated from the basis (the direction rows of the preceding and the present iterations) are marked by arrows.

* [Also called pivot, pivotal, determining, or key element.]

The principal part of the $(l+1)$-th tableau is filled in according to the recurrence formulas (1.12). First the r-th row of the $(l+1)$-th tableau is filled. According to the recurrence formula for the r-th row of the $(l+1)$-th tableau, the r-th row of the l-th tableau should be divided by the direction element x_{rk}:

$$x_{rj}^{(l+1)}=\frac{x_{rj}^{(l)}}{x_{rk}^{(l)}}; \qquad j=0,\ 1,\ 2,\ \ldots,\ n. \tag{2.1}$$

To obtain the entries of the i-th row of the $(l+1)$-th tableau ($i=1,\ 2, \ldots,\ m,\ m+1;\ i\neq r$), we subtract from the i-th row of the l-th tableau the r-th row of the $(l+1)$-th tableau multiplied by $x_{ik}^{(l)}$:

$$x_{ij}^{(l+1)}=x_{ij}^{(l)}-\frac{x_{rj}^{(l)}}{x_{rk}^{(l)}}\,x_{ik}^{(l)}=x_{ij}^{(l)}-x_{rj}^{(l+1)}\,x_{ik}^{(l)}, \tag{2.2}$$

$$i=1,\ 2,\ \ldots,\ m,\ m+1;\ i\neq r;\ j=0,\ 1,\ 2,\ \ldots,\ n.$$

Having written all the parameters in the principal part of the $(l+1)$-th tableau, the iteration terminates. Proceeding from the principal part of the $(l+1)$-th tableau, we carry out the $(l+2)$-th iteration according to the same rules used in the $(l+1)$-th iteration based on the l-th tableau. Considerations of § 2 of the preceding chapter ensure that the iterative process is terminated either when an optimal program is reached or when unsolvability of the problem is established.

All the tableaus, except one, of the algorithm are filled according to the same rules. The exception is the zeroeth tableau with which the computations start. The zeroeth tableau contains an additional row C in which the coefficients c_j of the linear form are written. The initial basis and the basis components of the initial support program are assumed to be given. They are written in the corresponding positions of the columns B_X and $A_0=B$. The entries x_{ij} ($i=1, 2, \ldots, m;\ j=1,\ 2,\ \ldots,\ n$) are also assumed to be known. If the initial basis consists of unit vectors, $x_{ij}=a_{ij},\ x_{i0}=b_i$. In more complicated cases, various artificial methods for computing x_{ij} are used. In some instances the x_{ij} must be computed from systems of linear equations.

If the initial program is not given in advance, then, applying the techniques of § 8, we may, simultaneously, determine the initial program and all the relevant parameters to be written in the zeroeth tableau.

The entries of the $(m+1)$-th row of the zeroeth tableau are computed from

$$x_{m+1,\,j}=\sum_{i=1}^{m}x_{ij}c_{s_i}-c_j, \qquad j=0,\ 1,\ 2,\ \ldots,\ n. \tag{2.3}$$

(We recall that $c_0=0$ and

$$x_{m+1,\,0}=\sum_{i=1}^{m}x_{i0}c_{s_i}=L\,(X).)$$

Thus, to obtain the $(m+1)$-th row of the zeroeth tableau we subtract row C (the first row of the tableau) from the row product of columns A_j and column C_X (the first column on the left of the tableau).

The principal part of the zeroeth tableau contains all the initial data required for carrying out the first iteration. Formulas (2.3) can, obviously, be used for direct computation of the entries in the last row of any tableau of the first algorithm.

Since $x_{m+1,j}$ can be computed by two independent methods we have a means of control in each iteration. After a certain number of iterations it is always advisable to compare the entries in the $(m+1)$-th row of the tableau as obtained from the recurrence formulas with the values of the linear form and the evaluations Δ_j as computed directly from formulas (2.3). If the values of the linear form obtained by the two methods do not coincide, we first must check whether the entries of the column A_0 constitute a feasible program of the problem. If x_{i0} do not satisfy the problem restraints, it is advisable to repeat all the computations starting with the iteration for which the previous check was made. Assume that the values of the linear form obtained by the two methods coincide, but the evaluations Δ_j are different for some j. In this case, the corresponding restraint vectors A_j should be expanded anew in terms of the basis vectors. Several systems of linear equations differing only in the right-hand sides are conveniently solved by the Gauss method (see Appendix, 2-7 and /110/).

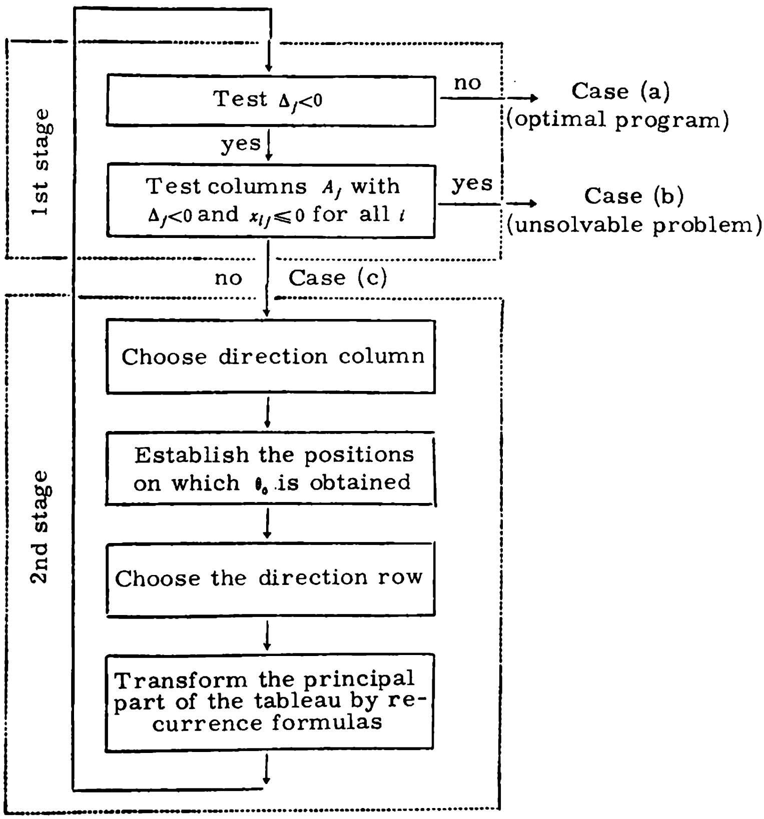


FIGURE 5.1

In Figure 5.1 we give a block diagram of a single iteration for the solution of a linear-programming problem by the first simplex algorithm.

2-3. To complete the discussion on the first algorithm, we now consider the rule applied to guarantee against cycling arising in degenerate linear-

programming problems. In computations according to the first algorithm, the basis vectors A_{s_i} of the support program (it is assumed that the enumeration of the restraint vectors is preserved in computations) are, obviously, chosen for the vectors $R_1, R_2, \ldots, R_m$ (see § 6 of the preceding chapter). The basis vectors satisfy the two conditions imposed on the system $(R_1, \ldots, \ldots, R_m)$. First, the vectors A_{s_i} are linearly independent, second, the basis of the support program X is also the basis of some program of the ε-problem where

$$B(\varepsilon)=B+\sum_{i=1}^{m} \varepsilon^i A_{s_i}.$$

The components of the support program of the ε-problem, i.e., the coefficients of $B(\varepsilon)$ expanded in terms of the basis vectors of program X, are equal to

$$x_{i_0}(\varepsilon)=x_{i_0}+\varepsilon^i.$$

Hence, the rule proposed in Chapter 5, § 6, for preventing cycling may be applied with $R_i=A_{s_i}$. The basis vectors of the support program X are chosen for the system $\{R_i\}$ because during the process of solution of the problem according to the first algorithm these vectors will always have to be expanded in the current basis.

To apply the rule for avoiding cycling, columns θ', θ'', ... are added on the right to the iteration tableaus corresponding to degenerate programs. In the degenerate case

$$\theta_0=\min_{x_{ik}>0} \frac{x_{i0}}{x_{ik}}$$

is obtained simultaneously on several positions. We take $R_1=A_{s_1}$ and write in column θ' the ratio $\frac{x_{is_1}}{x_{ik}}$ for the i-values on which θ_0 is obtained. We choose θ_0', the least of these numbers. The vector on which θ_0' is obtained should be eliminated from the basis. If there are several such vectors, the aforementioned procedure is repeated. In column θ'' we write the ratio $\frac{x_{is_2}}{x_{ik}}$ for the i-values on which θ_0' is obtained. Computations are continued until the position of the vector to be eliminated from the basis is uniquely specified.

2-4. We now estimate the bulkiness of computations according to the first simplex algorithm. The computer time required to solve the problem is determined, mainly, by the amount of multiplications and divisions to be carried out in the process of optimal-program determination. The operations of comparison, addition, and subtraction in universal computers are much faster than multiplication and division. The memory-address time is neglected here. Using the first algorithm, division is necessary twice in each iteration: when filling the tableau row corresponding to the new basis vector $\left(x_{rj}'=\frac{x_{rj}}{x_{rk}}, j=0, 1, \ldots, n\right)$ and when the column θ is computed $\left(\theta=\frac{x_{i0}}{x_{ik}} \text{ with } x_{ik}>0\right)$. Computation of the entries of the x_{rj}'-th row requires $n-m+1$ divisions, and computation of the elements of the column θ terminate after at most m divisions. The total number of divisions in each iteration of the first algorithm is, thus, at most $n+1$. Multiplication is necessary in the first algorithm only when transforming the tableaus according to the recurrence formulas (2.2):

$$x_{ij}^{(l+1)}=x_{ij}^{(l)}-x_{rj}^{(l+1)}x_{ik}^{(l)}; \qquad i=1, 2, \ldots, m, m+1;$$
$$i\neq r; \quad j=0, 1, 2, \ldots, n.$$

The coefficients $x_{ij}^{(l+1)}$ corresponding to the basis vectors are not computed here (they are either zero or one). In each iteration there are, therefore, $(n-m+1)m$ multiplications.

The number of iterations necessary to solve the linear-programming problem varies depending on the problem under consideration and the initial support program. There are no standard theoretical procedures for estimating the number of iterations involved in solving a general linear-programming problem. Experience shows that, as a rule, the number of iterations in various linear-programming problems ranges from m to $2m$

We repeat that for control purposes it is advisable, when computing according to the first algorithm, to compute the evaluations Δ_j after a certain number of iterations not only from the recurrence formulas, but also directly from (2.3). This requires $m(n+1)$ multiplications. If the evaluations of the basis vectors are not checked ($\Delta_j=0$ for $i \in I_x$), the number of additional multiplications necessary for control purposes is $m(n-m+1)$, which is equal to the number of multiplications in a single iteration. The control procedure under the first algorithm is, thus, fairly bulky. We know of no estimates for the optimum frequency for carrying out the computational check. Detailed study of this problem, at least for particular classes of problems, is obviously of considerable practical interest.

§ 3. Examples

We shall illustrate the sequence of computations according to the first algorithm by two examples.

Example 1. Maximize the linear form

$$L(X)=3x_1-x_2+8x_3+2x_4-x_5+9x_6$$

subject to the conditions

$$\begin{aligned}
-6x_1+9x_2+3x_3 \qquad -2x_5-x_6 &\leq 12,\\
-4x_2+3x_3-3x_4+x_5-x_6 &\leq 5,\\
2x_1+8x_2-5x_3+6x_4-8x_5+4x_6 &\leq 20,\\
-x_1-3x_2-4x_3-8x_4 \qquad +4x_6 &\leq 10,\\
5x_1+x_2+2x_3+4x_4+9x_5+5x_6 &\leq 24,\\
x_j \geq 0, \quad j=1, 2, \ldots, 6.&
\end{aligned}$$

To reduce the problem to canonical form, we introduce five additional nonnegative variables $x_7, \ldots, x_{11}$. Unit restraint vectors correspond to these new variables. The unit vectors $A_7, \ldots, A_{11}$ should, obviously, be taken as the initial basis. The entire solution process is given in Tables 5.2. The tables consist of tableaus corresponding to the separate iterations of the simplex procedure, and they are numbered according to the sequence of iterations.

The principal part of the zeroeth tableau (except for its first and last rows) is filled with the corresponding constraint vector (A_0) and restraint vectors ($A_1, A_2, \ldots, A_{11}$). The components of the constraint vector are written in the tableau as the basis components of the initial support program, and the components of the restraint vectors as the coefficients of the vectors A_j expressed in terms of the unit basis vectors.

The column C_X remains empty, since zero coefficients correspond to the additional variables in the linear form.

The last row of the zeroeth tableau contains the evaluations Δ_j of the restraint vectors. The parameters $\Delta_j=x_{m+1,j}$ are computed from (2.3). In our case, $c_{s_i}=0$. Therefore, in the Δ_j-th row, we write the elements of the first row (C) with reversed sign. The linear form vanishes on the initial program.

Among the evaluations Δ_j of the restraint vectors there are negative numbers. The initial support program is, therefore, not optimal. In the expansion of each of the restraint vector with negative evaluations there are positive x_{ij}. Hence, there is no reason to assume that the problem is unsolvable. We thus have case (c).

The least evaluations $\Delta_j=-9$ corresponds to A_6. The vector A_6 should be introduced into the basis. The last three elements (x_{36}, x_{46}, x_{56}) of the direction column A_6 are positive. The θ-values are therefore

computed only for the third, fourth, and fifth entries of the column θ. The least entry in the last column corresponds to the fourth basis position, being equal to

$$\theta_0 = \frac{x_{40}}{x_{46}} = \frac{10}{4} = 2.5.$$

This indicates that the fourth row is the direction row of the transformation and that the vector A_{10} situated in this row is to be eliminated from the basis.

We now proceed to compile the 1st tableau corresponding to the first iteration. In the column B_X the fourth entry is A_6, which has replaced A_{10} in the fourth position of the basis. The corresponding coefficient c_6 of the linear form is written in column C_X. The principal part of the 1st tableau is filled from the data of the preceding zeroeth tableau according to the recurrence formulas

$$x'_{ij} = x_{ij} - \frac{x_{4j}}{x_{46}} x_{i6}, \qquad i = 0, 1, 2, 3, 5, 6; \quad j = 0, 1, 2, \ldots, 11; \tag{3.1}$$

$$x'_{4j} = \frac{x_{4j}}{x_{46}}; \qquad j = 0, 1, 2, \ldots, 11. \tag{3.2}$$

The direction element is $x_{46} = 4$. Hence, the fourth row of the 1st tableau is obtained from the fourth row of the zeroeth tableau when the latter is divided by 4. The remaining entries in the principal part of the 1st tableau, the components of the support program and the evaluations of the restraint vector included, are computed from recurrence formulas (3.1). For instance,

$$x'_{23} = x_{23} - \frac{x_{43}}{x_{46}} x_{26} = 3 - \frac{-4(-1)}{4} = 2,$$

$$x'_6 = x'_{20} = x_{20} - \frac{x_{40}}{x_{46}} x_{26} = 5 - \frac{10(-1)}{4} = 7.5,$$

$$\Delta'_2 = x'_{02} = x_{02} - \frac{x_{42}}{x_{46}} x_{06} = 1 - \frac{-3(-9)}{4} = -5.75.$$

The iteration terminates when the principal part of the tableau is filled. The next iteration is carried out following the same rules. In each tableau arrows indicate the vectors introduced into the basis in the preceding iteration and the vectors to be eliminated from the basis. In the 7th tableau all the evaluations Δ_j are nonnegative. This indicates that the program $X^{(7)}$ is a solution of the problem.

To check the computations, it is advisable, at some stages, to determine the estimates Δ_j not only from the recurrence formulas (2.2), but also directly from (2.3). Let us compute, e. g., Δ'_2 in the 1st tableau:

$$\Delta'_2 = \sum_{i=1}^{6} x'_{i2} c_{s_i} - c_2 = 8.25.0 - 4.75.0 + 11.0 - 0.75.9 + 4.75.0 - (-1) = -5.75.$$

The same result was obtained from the recurrence formula (3.1).

Example 2. Tables 5.3 give the sequence of operations leading to the solution of a problem requiring minimization of the linear form

$$\tilde{L}(X) = -9x_1 + 8x_3 - 5x_5 + 3x_6 - 8x_8 - 8x_9,$$

subject to the conditions

$$\begin{array}{l}
6x_1 + 2x_3 + 2x_5 + 3x_8 \leq 24, \\
2x_2 + 3x_4 + 5x_7 + 9x_8 \leq 30, \\
9x_1 + x_2 + 5x_5 + x_6 \leq 40, \\
6x_3 + 2x_7 + x_9 \leq 36, \\
4x_4 + 8x_8 \leq 20, \\
3x_1 + x_3 + 8x_6 + 4x_7 + 4x_9 \leq 48, \\
x_j \geq 0, \quad j = 1, 2, \ldots, 9.
\end{array}$$

This problem is equivalent to maximization of the linear form

$$L(X) = 9x_1 - 8x_3 + 5x_5 - 3x_6 + 8x_8 + 8x_9$$

subject to the same conditions.

Tables 5.3 do not call for special explanations. The solution is obtained in five iterations (the zeroeth iteration is not considered as an independent step; the principal part of the zeroeth tableau contains data preparatory for the first iteration).

2154

TABLE 5.2 (0–7)

C	—	—	—	—	3	−1	8	2	−1	9						—	
	No.	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	θ	No. of tableau
	1		A_7	12	−6	9	3		−2	−1	1					—	0
	2		A_8	5		−4	3	−3	1	−1		1				—	
	3		A_9	20	2	8	−5	6	−8	4			1			5	
←	4		A_{10}	10	−1	−3	−4	−8		4				1		2.5	
	5		A_{11}	24	5	1	2	4	9	5					1	4.8	
	6	—	—		−3	1	−8	−2	1	−9						—	
	1		A_7	14.5	−6.25	8.25	2	−2	−2		1			0.25		—	1
	2		A_8	7.5	−0.25	−4.75	2	−5	1			1		0.25		—	
←	3		A_9	10.0	3.00	11.00	−1	14	−8				1	−1.00		0.714	
→	4	9	A_6	2.5	−0.25	−0.75	−1	−2		1				0,25		—	
	5		A_{11}	11.5	6.25	4.75	7	14	9					−1,25	1	0.822	
	6	—	—	22.5	−5.25	−5.75	−17	−20	1					2.25		—	
	1		A_7	15.929	−5.821	9.821	1.857		−3.143		1		0.143	0.107		8.577	2
	2		A_8	11.071	0.821	−0.821	1.643		−1.857			1	0.357	−0.107		6.739	
→	3	2	A_4	0.714	0.214	0.786	−0.071	1	−0.571				0.071	−0.071		—	
	4	9	A_6	3.929	0.179	0.821	−1.143		−1.143	1			0.143	0.107		—	
←	5		A_{11}	1.500	3.250	−6.250	8.000		17.000				−1.000	−0.250	1	0.188	
	6	—	—	36.786	−0.964	9.964	−18.429		−10.429				1.429	0.821		—	

TABLE 5.2 (continued)

	C	—	—	—	—	3	−1	8	2	−1	9						—	
	No	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	θ	No. of tableau
	1		A_7	15.580	−6.576	11.272			−7.089		1		0.375	0.165	−0.232	1.382	
	2		A_8	10.736	0.154	0.462			−5.348				0.562	−0.056	−0.205	23.295	
←	3	2	A_4	0.728	0.243	0.730		1	−0.420				0.062	−0.074	0.009	0.997	3
	4	9	A_6	4.143	0.643	−0.071			1.286	1				0.071	0.143	—	
→	5	8	A_3	0.188	0.406	−0.781	1		2.125				−0.125	−0.031	0.125	—	
	6	—	—	40.241	6.522	−4.433			28.732				−0.875	0.246	2.304	—	
	1		A_7	4.343	−10.333			−15.443	−0.609		1		−0.590	1.303	0.370		
	2		A_8	10.303				−0.633	−5.803			1	0.523	−0.009	−0.211	19.702	
⇄	3	−1	A_2	0.997	0.333	1		1.370	−0.575				0.086	−0.101	0.012	11.643	4
	4	9	A_6	4.214	0.667			0.098	1.245	1			0.006	0.064	0.144	689.0	
	5	8	A_3	0.966	0.667		1	1.070	1.676				−0.058	−0.110	0.135	—	
	6	—	—	44.661	8.000			6.073	26.183				−0.495	−0.202	2.358	—	
	1		A_7	11.214	−8.036	6.893		−6	−4.571		1			0.607	−0.286	18.471	
←	2		A_8	4.214	−2.036	−6.107		−9	−1.571			1		0.607	−0.286	6.941	
→	3		A_9	11.643	3.893	11.697		16	−6.714				1	−1.179	0.143	—	5
	4	9	A_6	4.143	0.643	−0.071			1.286	1				0.071	0.143	58.000	
	5	8	A_3	1.643	0.893	0.679	1	2	1.286					−0.179	0.143	—	
	6	—	—	50.429	9.929	5 786		14	22.857					−0.786	2.429	—	

TABLE 5.2 (continued)

C	—	—	—	—	3	−1	8	2	−1	9						—	No. of tableau
	No.	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	θ	
←	1		A_7	7.000	−6.000	13.000		3.000	−3.000		1	−1				0.538	6
→	2		A_{10}	6.941	−3.353	−10.059		−14.824	−2.588			1.647		1	−0.471	—	
	3		A_9	19.824	−0.059	−0.176		−1.471	−9.765			1.941	1		−0.412	—	
	4	9	A_6	3.647	0.882	0.647		1.059	1.471	1		−0.118			0.176	5.636	
	5	8	A_3	2.882	0.294	−1.118	1	−0.647	0.824			0.294			0.059	—	
	6	—	—	55.882	7.294	−2.118		2.353	20.824			1.294			2.059	—	
→	1	−1	A_2	0.538	−0.462	1		0.231	−0.231		0.077	−0.077					7
	2		A_{10}	12.357	−7.995			−12.502	−4.910		0.774	0.873		1	−0.471		
	3		A_9	19.919	−0.140			−1.430	−9.805		0.014	1.928	1		−0.412		
	4	9	A_6	3.299	1.181			0.910	1.620	1	−0.050	−0.068			0.176		
	5	8	A_3	3.484	−0.223		1	−0.389	0.566		0.086	0.208			0.059		
	6	—	—	57.023	0.632			2.842	20.335		0.163	1.131			2.059	—	

C	—	—	—	—	9		−8		5	−3
	No.	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5	A_6
←	1		A_{10}	24	6		2		2	
	2		A_{11}	30		2		3		
	3		A_{12}	40	9	1			5	1
	4		A_{13}	36			6			
	5		A_{14}	20				4		
	6		A_{15}	48	3		1			8
	7	—	—		−9		8		−5	3
→	1	9	A_1	4	1		0.333		0.333	
	2		A_{11}	30		2		3		
	3		A_{12}	4		1	−3		2	1
	4		A_{13}	36			6			
	5		A_{14}	20				4		
←	6		A_{15}	36					−1	8
	7	—	—	36			11		−2	3
	1	9	A_1	4	1		0.333		0.333	
	2		A_{11}	30		2		3		
	3		A_{12}	4		1	−3		2	1
	4		A_{13}	27			6		0.25	−2
←	5		A_{14}	20				4		
→	6	8	A_9	9					−0.25	2
	7	—	—	108			11		−4	19

TABLE 5.3 (0–5)

	8	8							—	
A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	θ	No. of tableau
	3		1						4	
5	9			1					—	
					1				4,444	
2		1				1			—	0
	8						1		—	
4		4						1	16	
	−8	−8							—	
	0.5		0.167						—	
5	9			1					—	
	−4.5		−1.5		1				—	
2		1				1			36	1
	8						1		—	
4	−1.5	4	−0.5					1	9	
	−3.5	−8	1.5						—	
	0.5		0.167						8	
5	9			1					3.333	
	−4.5		−1.5		1				—	
1	0.375		0.125			1		−0.25	72	2
	8						1		2.5	
1	−0.375	1	−0.125					0.25	—	
8	−6.5		0.5					2	—	

C	—	—	—	—	9		−8		5
	No.	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5
	1	9	A_1	2.75	1		0.333	−0.25	0.333
	2		A_{11}	7.5		2		−1.5	
←	3		A_{12}	15.25		1	−3	2.25	2
	4		A_{13}	26.062			6	−0.188	0.25
→	5	8	A_8	2.50				0,5	
	6	8	A_9	9.938				0.188	−0.25
	7	—	—	124.25			11	3.25	−4
←	1	9	A_1	0.208	1	−0.167	0.833	−0.625	
	2		A_{11}	7.5		2		−1.5	
→	3	5	A_5	7.625		0.5	−1.5	1.125	1
	4		A_{13}	24.156		−0.125	6.375	−0.469	
	5	8	A_8	2.5				0.5	
	6	8	A_9	11.844		0.125	−0.375	0.469	
	7	—	—	154.75		2	5	7.75	
→	1		A_{10}	0.5	2.4	−0.4	2	−1.5	
	2		A_{11}	7.5		2		−1.5	
	3	5	A_5	8	1.8	0.2			1
	4		A_{13}	24	−0.75		5.75		
	5	8	A_8	2.5				0.5	
	6	8	A_9	12	0,75		0.25		
	7	—	—	156	6	1	10	4	

TABLE 5.3 (continued)

−3		8	8							—	No. of tableau
A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	θ	
				0.167				−0.062		8.25	
	5				1			−1.125		—	
1				−1.5		1		0.562		7.625	
−2	1			0.125			1	−0.047	−0.25	104.25	3
		1						0.125		—	
2	1		1	−0.125				0.047	0.25	—	
19	8			0.5				0.812	2	—	
−0.167				0.417		−0.167		−0.156		0.5	
	5				1			−1.125		—	
0.5				−0.75		0.5		0.281		—	
−2.125	1			0.312		−0.125	1	−0.117	−0.25	77.3	4
		1						0,125		—	
2.125	1		1	−0.312		0.125		0.117	0.25	—	
21	8			−2.5		2		1.938	2	—	
−0.4				1		−0.4		−0.375			
	5				1			−1.125			
0.2						0.2					
−2	1						1		−0.25		5
		1						0.125			
2	1		1						0.25		
20	8					1		1	2	—	

§ 4. The coordinate form of the simplex method

4-1. In the preceding chapter, in our discussion of the simplex method, we started with such concepts as restraint vector, constraint vector, support program basis. The transformation from one support program to the successive one was effected by replacing one of the basis vectors by another, and the simplex parameters in each iteration were determined by expanding the restraint vectors and the constraint vector in terms of the basis vectors. The name "vector form" suggests itself for describing this approach to the simplex method.

There are also other approaches to the description of the method and construction of the corresponding algorithms. In particular, the structure of the tableaus of the first algorithm is directly determined by the so-called coordinate form of the simplex method. An analysis of the coordinate form cannot contribute anything essentially new to the fundamentals of the method. Nonetheless, a description of a new form of the method, with due emphasis on the individual details, produces fresh analogies and is thus conducive to the assimilation of both the theoretical principles of the method and the computational procedures.

Consider the linear-programming problem in canonical form.

Maximize the linear form

$$L(X)=\sum_{j=1}^{n} c_j x_j \tag{4.1}$$

subject to the conditions

$$\sum_{j=1}^{n} A_j x_j = B, \tag{4.2}$$

$$x_j \geqslant 0, \quad j=1,\ 2,\ \dots,\ n. \tag{4.3}$$

Let $X^0=(x_1^0,\ \dots,\ x_n^0)$ be some support program of problem (4.1)–(4.3) with basis $A_{s_1},\ \dots,\ A_{s_m}$; let the set of indices $s_1,\ \dots,\ s_m$ be denoted by I_0, and the matrix of the basis vectors $A_j (j \in I_0)$ by A_{X^0}. Thus

$$A_{X^0}=(A_{s_1},\ \dots,\ A_{s_m}).$$

Let $X_j^0=(x_{1j}^0,\ \dots,\ x_{mj}^0)^T$ be the vector whose components are the coefficients in the expansion of the vectors $A_j (j=0,\ 1,\ \dots,\ n)$ in terms of the basis of program X^0:

$$A_j = A_{X^0} X_j^0,$$

or, equivalently,

$$X_j^0 = A_{X^0}^{-1} A_j.$$

Multiplying restraints (4.2) on the left by $A_{X^0}^{-1}$, we obtain

$$\sum_{j=1}^{n} X_j^0 x_j = X_0^0. \tag{4.4}$$

The vectors $X_{s_i}^0$ corresponding to the basis components of the support program X are unit vectors. Therefore, (4.4) can be rewritten as follows:

$$x_{s_i} + \sum_{j \notin I_0} x_{ij}^0 x_j = x_{s_i}^0, \qquad i=1,\ 2,\ \dots,\ m.$$

As usual, we establish the relation between the notations of the basis

variables of the support programs and the positions of the corresponding vectors in the basis. We have

$$x_{s_i}=x_{i0}^0-\sum_{j\notin I_0} x_{ij}^0 x_j, \qquad i=1,\ 2,\ \ldots,\ m. \tag{4.5}$$

Applying the above relationships, we rewrite (4.1) as follows:

$$\begin{aligned} L(X)&=\sum_{j=1}^{n} c_j x_j=\sum_{i=1}^{m} c_{s_i} x_{s_i}+\sum_{j\notin I_0} c_j x_j=\\ &=\sum_{i=1}^{m} c_{s_i}\left(x_{i0}^0-\sum_{j\notin I_0} x_{ij}^0 x_j\right)+\sum_{j\notin I_0} c_j x_j=\\ &=\sum_{i=1}^{m} c_{s_i} x_{i0}^0+\sum_{j\notin I_0} x_j\left(c_j-\sum_{i=1}^{m} c_{s_i} x_{ij}^0\right), \end{aligned}$$

whence

$$L(X)=L(X^0)-\sum_{j\notin I_0} \Delta_j^0 x_j. \tag{4.6}$$

Let

$$\left.\begin{aligned} L(X)=x_{s_{m+1}};\quad L(X^0)=x_{s_{m+1}}^0=x_{m+1,\,0}^0;\\ \Delta_j^0=x_{m+1,\,j}^0. \end{aligned}\right\} \tag{4.7}$$

In these new notations, relationships (4.5) and (4.6) are combined into a single system

$$x_{s_i}=x_{i0}^0-\sum_{j\notin I_0} x_{ij}^0 x_j, \quad i=1,\ 2,\ \ldots,\ m,\ m+1. \tag{4.8}$$

Formulas (4.8) give an alternative representation of the linear form (4.1) and restraints (4.2) of a problem where the restraint matrix and the coefficients of the linear form are replaced by the parameters of a given support program.

We now compare (4.8) with the zeroeth tableau of the first algorithm. We see that the parameters characterizing the initial support program are written in the $(m+1)$ rows of the tableau in the same order as they appear in equalities (4.8). The last $(m+1)$-th relationship of system (4.8) (in previous notations, formula (4.6)) shows that if all $\Delta_j^0\geqslant 0$, then for $x_j\geqslant 0$ $(j\notin I_0)$ the linear form $L(X)$ may not exceed $L(X^0)$; the support program X^0 proves to be the optimal program of the problem (case (a)). If for at least one $\Delta_j^0<0$ all $x_{ij}^0<0$ $(i=1,\ 2,\ \ldots,\ m)$, we see from (4.8) that when x_j increases to infinity programs X, on which $L(X)$ increases to infinity, are obtained. Thus the condition

$$x_{m+1,\,j}^0=\Delta_j^0<0$$

and

$$x_{ij}^0<0$$

indicate unsolvability of the problem (case (b)).

Now we assume that for all the indices j for which $\Delta_j^0<0$ there exists at least one $x_{ij}^0>0$ (case (c)). We shall show how to transfer to program X' at the same time increasing the value of the linear form.

Take $k\notin I_0$ such that

$$x_{m+1,\,k}^0=\Delta_k^0<0.$$

Now, express one of the x_k in (4.8) in terms of the other variables. Let the number, i, of the equation from which x_k is taken be r. Solving this equation for x_k we obtain

$$x_k=x_k^{\prime\prime}-\sum_{j\notin I_1} x_{rj}^{\prime} x_j,$$

where

$$x'_k = \frac{x^0_{r0}}{x^0_{rk}}, \quad x'_{kj} = \frac{x^0_{rj}}{x^0_{rk}},$$

and the set of indices I_1 is obtained from I_0 when k is substituted for s_r. Let $I_1 = \{s'_1, s'_2, \ldots, s'_m\}$ where $s'_i = s_i$ for $i \neq r$, and $s'_r = k$. Moreover, let

$$x'_k = x'_{s'_r} = x'_{r0}.$$

Then

$$x_{s'_r} = x'_{r0} - \sum_{j \in I_1} x'_{rj} x_j. \tag{4.9}$$

These new notations are justified since the variables x_k and x'_k are obtained from the r-th equation of system (4.8).

Substituting the expression for $x_k = x_{s'_r}$ into the other equations of system (4.8), we obtain

$$x_{s'_i} = x'_{i0} - \sum_{j \in I_1} x'_{ij} x_j, \qquad i = 1, 2, \ldots, m, m+1; \quad i \neq r, \tag{4.10}$$

where

$$x'_{ij} = x^0_{ij} - \frac{x^0_{rj}}{x^0_{rk}} x^0_{ik}.$$

Combining (4.9) and (4.10), we rewrite the transformed system (4.8) in the form

$$x_{s'_i} = x'_{i0} - \sum_{j \in I_1} x'_{ij} x_j, \tag{4.11}$$

where

$$x'_{ij} = \begin{cases} x^0_{ij} - \dfrac{x^0_{rj}}{x^0_{rk}} x^0_{ik} & \text{for } i \neq r, \\ \dfrac{x^0_{rj}}{x^0_{rk}} & \text{for } i = r, \end{cases} \tag{4.12}$$

$$i = 1, 2, \ldots, m+1; \; j = 0, 1, 2, \ldots, n.$$

Until now r was an arbitrary index. However, to ensure nonnegativity of the components x'_{i0} $(i = 1, \ldots, m)$, r must be chosen subject to the condition:

$$x'_{i0} \geqslant 0, \quad i = 1, 2, \ldots, m.$$

It follows from (4.12) that, for $x^0_{ik} \leqslant 0$, x'_{i0} are always positive. For $x^0_{ik} > 0$, x'_{i0} are positive if

$$\frac{x^0_{r0}}{x^0_{rk}} \leqslant \frac{x^0_{i0}}{x^0_{ik}}.$$

Hence r must be chosen subject to the condition:

$$\frac{x^0_{r0}}{x^0_{rk}} = \min_{\substack{1 \leqslant i \leqslant m \\ x_{ik} > 0}} \frac{x^0_{i0}}{x^0_{ik}}. \tag{4.13}$$

We now write (4.12) for $i = m+1$ and $j = 0$ in the previous notations:

$$L(X') = L(X^0) - \frac{x^0_{r0}}{x^0_{rk}} \Delta^0_k,$$

where X^0 and X' are support programs of problem (4.1)–(4.3) with the basis

components $x^0_{s_i}$ and $x'_{s'_i}$ $(i=1, 2, \ldots, m)$ respectively. For given k and r and $\frac{x^0_{r0}}{x^0_{rk}}=\theta_0>0$

$$L(X')>L(X^0).$$

To sum up: having suitably chosen the indices k and r, we applied the identity transformation to (4.8) thus obtaining (4.11) and then passed from program $X^0\geqslant 0$ to program $X'\geqslant 0$, in which $L(X')>L(X^0)$.

The structure of system (4.11), like that of system (4.8), is such that if the variables x_j over which summation is carried out (extrabasis variables) are taken equal to zero, we obtain a support program and the value of the linear form of problem (4.1)–(4.3).

Let us compare system (4.11) with the l-th tableau (Table 5.1) for $l=1$ of the computational procedure according to the first algorithm. We see, in the tableau, that the support-program parameters are written in the same order as in the right-hand sides of equations (4.11). The first-algorithm tableaus, in fact, reflect the successive transformations associated with the solution of a system of linear equations by the complete elimination method (the Gauss method). The only feature which converts the Gauss method of solution of a system of linear equations into a method of solution of the corresponding linear-programming problem lies in the special choice of the direction element x_{rk} (the choice of the indices r and k).

We assert that the simplex method actually specifies a process for solving a system of linear equations – the problem restraints – by the complete elimination method supplemented by a special rule for choosing the direction element. The latter ensures succession of programs (the selection rule for the row r) and monotonic increase of the linear form (the selection rule for the column k). As we shall see below, other finite methods of linear programming are also reducible to the complete elimination method, though modified by other selection rules for the direction element.

§ 5. The second simplex algorithm

5-1. From the second form of the optimality test we obtain a variation of the simplex procedure, the so-called second algorithm or the inverse-matrix method.

A variation of the simplex method, close to that described here, was first applied by L. V. Kantorovich to one of the particular problems of linear programming /66, 63/. Later (1951), this algorithm, as one of the realizations of the so-called method of decision multipliers, was applied to the solution of the general linear-programming problem /67/.

We shall consider the linear-programming problem in canonical form, problem (4.1)–(4.3). Let X be a support program of the problem with the basis $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$. Let I_X indicate the set of indices of the basis vectors (the numbers of basis variables).

We construct the square matrix A_X of order m. The columns of A_X are the basis restraint vectors. The determinant of A_X, therefore, does not vanish, and the inverse A_X^{-1} exists. We shall show that, using A_X^{-1} and the problem restraints, we obtain a fairly compact procedure for computing the parameters required in subsequent program improvements.

To use the simplex method, we must know, given an initial support program and its basis, how to choose in each iteration a restraint vector A_k to be introduced into the basis and a vector A_{s_r} to be eliminated from the basis. The vector A_k is a restraint vector with a negative relative evaluation Δ_k(generally, $\Delta_k = \min_j \Delta_j$). The vector A_{s_r} is the vector on which the ratio $\frac{x_{i0}}{x_{ik}}$, for $x_{ik} > 0$, obtains its least value. Therefore, to pass from a given program to another program ensuring a higher value of the linear form we must know how to compute in each iteration the basis components x_{i_0} of the support programs, the evaluations Δ_j of the restraint vectors with respect to the support-program basis, and the coefficients x_{ik} in the expansion of the vector A_k, to be introduced into the basis, in terms of the vectors of the current basis.

According to the first algorithm, to determine these parameters in each iteration we had to compute the coefficients x_{ij} of all the restraint vectors A_j written in terms of the basis vectors. As we shall see below, all the parameters necessary for program evaluation and for transferring to another, better program can be obtained by step-by-step transformation of the elements of the inverse matrix A_X^{-1}.

Indeed, the basis components of the support program are the elements of the product of matrix A_X^{-1} and constraint vector B. The evaluations Δ_j of the restraint vector can be computed (see Chapter 4, § 2) from

$$\Delta_j = z_j - c_j = \sum_{i=1}^{m} a_{ij}\lambda_i - c_j = (\Lambda, A_j) - c_j, \quad j = 1, 2, \ldots, n, \tag{5.1}$$

where the parameters λ_i for $j \in I_X$ satisfy the equations

$$\sum_{i=1}^{m} a_{ij}\lambda_j = c_j,$$

or, equivalently,

$$\Lambda A_X = C_X, \tag{5.2}$$

where C_X is a row vector comprising the linear-form coefficients which correspond to the basis variables. Hence,

$$\Lambda = C_X A_X^{-1}. \tag{5.3}$$

With the aid of (5.1) and (5.3) we can compute the evaluations Δ_j of the restraint vector from the elements of the inverse A_X^{-1} and the problem restraints.

The coefficients x_{ik} in the expansion of the vector A_k in terms of the current basis are the elements of the product of matrix A_X^{-1} and restraint vector A_k. The coefficients x_{ik} together with the basis components of the support program specify the vector to be eliminated from the basis. Moreover, the signs of the coefficients x_{ik} give an indication as to whether the problem is solvable. Obviously, with the second algorithm unsolvability of the problem, if it is indeed unsolvable, is usually detected later than with the first algorithm, where the signs of x_{ij} can be established for all the vectors A_j with negative evaluations.

Thus, to use the simplex method, it suffices to know how to compute in each iteration the inverse A_X^{-1} of the matrix constructed of the current basis vectors.

The column entries of the inverse A_X^{-1} are, conveniently, taken to be

the coefficients e_{ij} in the expansion of the unit vectors e_j $(j=1, 2, \ldots, m)$ in terms of the basis vectors. In § 1 we derived recurrence formulas (1.21) relating the parameters e_{ij} of two successive iterations. We recall that e_{ij} for $i=1, 2, \ldots, m$, $j=1, 2, \ldots, m$ are the coefficients in the expansion of the unit vectors in terms of basis vectors; $e_{i0}=x_{i0}$ are the basis variables of the support program; $e_{m+1,j}=\lambda_j$ are the evaluations of problem restraints with respect to the given program; $e_{m+1,0}=\lambda_0=L(X)$ is the value of the linear form.

Recurrence formulas (1.21) constitute the basis of the second simplex algorithm. The role played by the inverse matrix A_X^{-1} in the computational procedure justifies the name of the inverse-matrix method sometimes given to the second algorithm.

5-2. Below we give the sequence of computations under the second algorithm. Solution of the linear-programming problem according to the second simplex algorithm involves successively filling in a system of principal tableaus and an auxiliary tableau.

The principal tableaus have the following structure (see Table 5.4).

TABLE 5.4

The principal l-th tableau

No.	C_X	B_X	e_0	e_1	e_2	...	e_m	A_k	θ
1	c_{s_1}	A_{s_1}	$e_{10}^{(l)}$	$e_{11}^{(l)}$	$e_{12}^{(l)}$	...	$e_{1m}^{(l)}$	$x_{1k}^{(l)}$	
2	c_{s_2}	A_{s_2}	$e_{20}^{(l)}$	$e_{21}^{(l)}$	$e_{22}^{(l)}$	...	$e_{2m}^{(l)}$	$x_{2k}^{(l)}$	
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮
r	c_{s_r}	A_{s_r}	$e_{r0}^{(l)}$	$e_{r1}^{(l)}$	$e_{r2}^{(l)}$	...	$e_{rm}^{(l)}$	$x_{rk}^{(l)}$	θ_0
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮
m	c_{s_m}	A_{s_m}	$e_{m0}^{(l)}$	$e_{m1}^{(l)}$	$e_{m2}^{(l)}$	...	$e_{mm}^{(l)}$	$x_{mk}^{(l)}$	
$m+1$	—	—	$L^{(l)}$	$\lambda_1^{(l)}$	$\lambda_2^{(l)}$	...	$\lambda_m^{(l)}$	$\Delta_k^{(l)}$	

In the first column (No.) the number of the row is given. In the second column (C_X) the linear-form coefficients c_{s_i}, corresponding to the basis variables, are written. Under B_X we give the basis vectors A_{s_i}. The next $(m+1)$ columns constitute the principal part of the tableau which contains the coefficients $e_{ij}^{(l)}$ of the expansion of vectors e_j $(j=0, 1, \ldots, m)$ in terms of the basis vectors. Here $e_0=B$ is the constraint vector and e_j is an m-dimensional unit vector with unity in the j-th position. The $(m+1)$-th entry in each column in the principal part of the tableau is the value of $e_{m+1,j}^{(l)}$:

$$e_{m+1,j}^{(l)}=\begin{cases} L(X^{(l)}) \text{ for } j=0, \\ \lambda_j^{(l)} \quad \text{for } j=1, 2, \ldots, m. \end{cases}$$

On the right of the principal part of the tableau we have the column A_k in which the coefficients $x_{ik}^{(l)}$ of the expansion of the vector introduced into the basis in terms of the basis vectors is written. The $(m+1)$-th entry in the column A_k is the evaluation $\Delta_k^{(l)}$ of the vector A_k with respect to the basis. The entries in the last column of the principal tableau, column θ, are used for choosing the vector to be eliminated from the basis. The column θ is filled, as in the first algorithm, with the ratios of the basis components $e_{i0}^{(l)}$ of the program to the corresponding entries $x_{ik}^{(l)}$ in column A_k. The entries in column θ corresponding to $x_{ik}^{(l)} \leqslant 0$ are crossed out.

The upper part of the auxiliary tableau (Table 5.5) contains the augmented restraint vectors (the restraint matrix and the linear-form coefficients) and the constraint vector B.

TABLE 5.5
The auxiliary tableau

No.	B	A_1	A_2	...	A_k	...	A_n
1	b_1	a_{11}	a_{12}	...	a_{1k}	...	a_{1n}
2	b_2	a_{21}	a_{22}	...	a_{2k}	...	a_{2n}
⋮	⋮	⋮	⋮		⋮		⋮
m	b_m	a_{m1}	a_{m2}	...	a_{mk}	...	a_{mn}
$m+1$	—	c_1	c_2		c_k	...	c_n
0	—	Δ_1	Δ_2	...	Δ_k	...	Δ_n
1	—	Δ_1'	Δ_2'	...	Δ_k'	...	Δ_n'
2	—	Δ_1''	Δ_2''	...	Δ_k''	...	Δ_n''
⋮	⋮	⋮	⋮		⋮		⋮

In the lower part of the auxiliary tableau evaluations of the restraint vector with respect to each of the support-program bases obtained in the process of solution are written in the rows. The rows in the lower part of the auxiliary tableau are numbered according to the number of iterations (more precisely, by the number of the principal tableaus). In row 0 we write evaluations of the restraint vector Δ_j with respect to the basis of the initial support program. With each iteration a new row $\Delta_j^{(l)}$ is added to the auxiliary tableau. The solution process is terminated when a row of nonnegative elements appear in the auxiliary tableau, or when a column A_k with nonpositive $x_{ik}^{(l)}$ is obtained in a principal tableau.

Let us now consider the sequence of computations in one iteration.

Let the l-th iteration be completed. The l-th principal tableau (with the exception of the last two columns A_k and θ) and the $\Delta^{(l)}$ row of the auxiliary tableau are filled in. In the first stage of the $(l+1)$-iteration, the $\Delta^{(l)}$-th row of the auxiliary tableau is examined. If all $\Delta_j^{(l)} \geqslant 0$, the support program obtained in the l-th iteration is optimal (case (a)). We now assume that there are vectors with negative evaluations. The vector A_k with the least evaluation $\Delta_k^{(l)}$ is introduced into the basis, and the coefficients $x_{ik}^{(l)}$ in the expansion of A_k in terms of the basis vectors are written in column A_k of the l-th principal tableau. Each $x_{ik}^{(l)}$ is obtained by multiplying column A_k in the upper part of the auxiliary tableau (positions $1, 2, \ldots, m$) by the i-th row of the l-th principal tableau ($j=1, 2, \ldots, m$):

$$x_{ik}^{(l)} = \sum_{j=1}^{m} e_{ij}^{(l)} a_{jk}. \tag{5.4}$$

The $(m+1)$-th entry in column A_k is the evaluation $\Delta_k^{(l)}$ of this vector.

Examine the column A_k. If all $x_{ik}^{(l)} \leqslant 0$ ($i=1, 2, \ldots, m$) the problem is unsolvable (case (b)).

We continue with the second stage of the iteration if there exists at least one positive coefficient $x_{ik}^{(l)}$. (Observe that under the first algorithm the vector A_k to be introduced into the basis was chosen in the second stage of the iteration.)

In the second stage we determine the vector to be eliminated from the basis, compute the new support program and the initial data for the next iteration. In column θ, as in the first algorithm, the ratios of the basis components of the support program (elements of column e_0) to the corresponding components $x_{ik}^{(l)}$ of column A_k are written. The entries in column θ corresponding to $x_{ik}^{(l)} \leqslant 0$ are crossed out. The least element of column θ is denoted by θ_0. The basis vector A_{s_r} on which θ_0 is obtained should be eliminated from the basis. If program $X^{(l)}$ is degenerate and θ_0 is obtained on several vectors, any of these can be eliminated from the basis, e.g., the vector with the least index. If cycling occurs, the rule given in 5-5 should be applied.

The row r of the l-th principal tableau is singled out. The element $x_{rk}^{(l)}$ at the intersection of the r-th row (the d i r e c t i o n r o w) and the column A_k (the d i r e c t i o n c o l u m n) of the principal tableau is called the d i r e c t i o n e l e m e n t of the transformation of the l-th principal tableau into the $(l+1)$-th tableau.

In the r-th cell of column B_X of the $(l+1)$-th principal tableau we write the vector A_k. According to its position in the basis, the vector A_{s_r} is denoted as A_k in the $(l+1)$-th tableau. In the rest of the positions of the column B_X the same vectors as in the l-th tableau are written.

The principal part of the $(l+1)$-th tableau is filled by using the data of the l-th tableau and recurrence formulas (1.21). To obtain the r-th row in the principal part of the $(l+1)$-th tableau, we divide the r-th row of the l-th tableau by the direction element $x_{rk}^{(l)}$:

$$e_{rj}^{(l+1)} = \frac{e_{rj}^{(l)}}{x_{rk}^{(l)}}, \qquad j=0, 1, 2, \ldots, m. \tag{5.5}$$

To fill the i-th row of the principal part of the $(l+1)$-th tableau ($i=1, 2, \ldots, \ldots, m+1$; $i \neq r$), we subtract, according to the recurrence formulas, the r-th row of the $(l+1)$-th tableau multiplied by $x_{ik}^{(l)}$ from the i-th row of the

l-th tableau:

$$e_{ij}^{(l+1)}=e_{ij}^{(l)}-\frac{e_{rj}^{(l)}}{x_{rk}^{(l)}}x_{ik}^{(l)}=e_{ij}^{(l)}-e_{rj}^{(l+1)}x_{ik}^{(l)},$$
$$i=1,\ 2,\ \ldots,\ m,\ m+1;\ j=0,\ 1,\ 2,\ \ldots,\ m. \tag{5.6}$$

To complete the $(l+1)$-th iteration, the evaluations of the restraint vector with respect to the basis of the new support program, must be written in the $\Delta^{(l+1)}$-th row in the auxiliary tableau.

The $\Delta^{(l+1)}$-th row is filled according to formula (5.1). The linear-form coefficients c_j are written in the $(m+1)$-th row of the auxiliary tableau. The components a_{ij} of the vector A_j occupy the first m positions of the column A_j in the auxiliary tableau, and the components λ_j $(j=1, 2, \ldots, m)$ of the vector Λ fill the last row in the $(l+1)$-th principal tableau. Thus, to obtain the $\Delta^{(l+1)}$-th row we must subtract the row (C) of the auxiliary tableau from the product of matrix A (the upper part of the auxiliary tableau) and the last row of the $(l+1)$-th principal tableau.

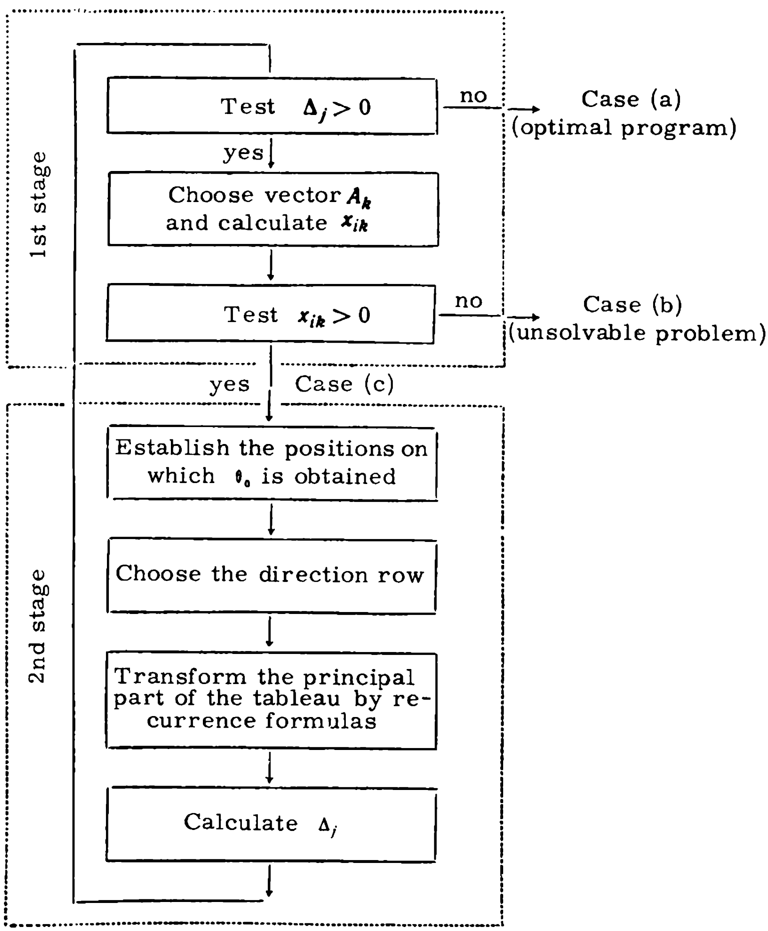


FIGURE 5.2

The $\Delta^{(l+1)}$-th row of the auxiliary tableau and the principal part of the $(l+1)$-th tableau contain all the initial data for the $(l+2)$-th iteration. Subsequent iterations follow the same outline. As shown in the discussion on the theoretical principles of the method, a finite number of iterations will produce an optimal program or establish unsolvability of the problem.

All the tableaus according to this algorithm are filled following the same rules. Some exceptions arise in compiling the zeroeth principal tableau. In column e_0 we write the basis components of the initial support program. The first m entries in columns $e_1, e_2, \ldots, e_m$ are obtained by direct inversion of the basis-vector matrix of the initial support program. The last entries in columns $e_1, \ldots, e_m$ are computed from formula (5.3). (We recall that the columns of matrix A_X^{-1} are the columns e_j of the zeroeth tableau.) Thus, the last row of the principal part of the zeroeth tableau is filled with the products of column C_X and columns e_j $(j=0, 1, \ldots, m)$ of this tableau.

Figure 5.2 shows a block diagram of a single iteration of the second simplex algorithm.

In order to check the computations carried out under the second algorithm, we should, obviously, use the two methods for determining λ_j. In any iteration, the parameters λ_j can be computed directly (as in the zeroeth tableau) or from recurrence formulas (1.21). All that has been said on the technique of control in connection with the first algorithm applies in this case too.

5-3. We now estimate the bulkiness of computations in one iteration of the second simplex algorithm. We have already observed that this aspect of the computational procedure should be evaluated in terms of the number of divisions and multiplications necessary in each iteration.

Under the second algorithm, division is necessary twice in each iteration – when computing the row in the new principal tableau corresponding to the vector newly introduced into the bases (see (5.5)) and when computing the column θ. In the first case there are $m+1$ divisions, and in the second there are at most m. Multiplication under the second algorithm is necessary three times: first, when transforming the principal tableau according to formulas (5.6), second when computing the evaluations Δ_j from (5.1), and third when determining the coefficients x_{ik} in the expansion of the vector A_k introduced into the basis in terms of the basis vectors. The transformation of the principal part of the tableau requires $m(m+1)$ multiplications (in formulas (5.6) $i=1, 2, \ldots, m, m+1$; $i \neq r$; $j=0, 1, \ldots, m$). The computation of Δ_j for each restraint vector A_j involves m multiplications and there is a total of $n-m$ evaluations of the restraint vector to be computed (as the number of extrabasis variables; the evaluations of the basis variables need not be computed, since they are all zero). Hence, computation of all the evaluations in one iteration involves $m(n-m)$ multiplications. Finally, to compute all the x_{ik} $(j=1, 2, \ldots, m)$ from formula (5.4), another m^2 multiplications must be carried out. We thus see that each iteration under the second algorithm involves at most $2m+1$ divisions and

$$m(m+1)+m(n-m)+m^2=m(n+m+1)$$

multiplications. To check computations, it is advisable to compute the parameters λ_j, at fixed intervals, not only from the recurrence formulas, but also directly from formulas (5.3). Each control stage thus involves m^2 additional multiplications.

Comparing the bulkiness of the individual stages in each iteration, it is

necessary to make the following comments concerning the choice of the direction element.

For $n \gg m$, the computation of evaluations becomes the most lengthy operation under the second algorithm. All the evaluations must be computed so that the least Δ_j can be chosen. The vector A_k for which

$$\Delta_k = \min_j \Delta_j$$

is introduced into the basis. There will, obviously, be far less operations if the evaluations are computed until the first negative Δ_j is obtained, and then the vector corresponding to this evaluation introduced into the basis. The number of iterations may increase; however, this is, generally, more than compensated for because the bulkiness of computations in each iteration is reduced. In each iteration the order in which the evaluations are computed is, generally speaking, arbitrary. We suggest the following sequence. For the initial program, compute the evaluations $\Delta_1, \ldots, \Delta_{p_1}$, where Δ_{p_1} is the first negative evaluation. For the next support program, compute $\Delta'_{p_1+1}, \ldots, \Delta'_{p_2}$, where Δ'_{p_2} again is the first negative number. If all $\Delta'_q \geqslant 0$ for $q = p_1+1, p_1+2, \ldots, n$, proceed with the computation of $\Delta'_1, \Delta'_2, \ldots$

5-4. The working memory is the bottleneck of modern computers. We shall outline a modification of the second algorithm, which enables the memory storage space to be used more economically /51/.

The preceding computational procedure requires that in each iteration the entire inverse matrix $A_X^{-1} = \|e_{ij}\|$ be stored in the memory. The so-called product form of the second algorithm (or product form for the inverse in the simplex method) involves storage of much less data.

The product form is based on the following considerations. Let X and X' be two successive support programs of the problem. Let the corresponding matrices of the basis vectors be A_X and $A_{X'}$:

$$A_X = (A_{s_1}, \ldots, A_{s_r}, \ldots, A_{s_m}), \quad A_{X'} = (A_{s_1}, \ldots, A_k, \ldots, A_{s_m}).$$

It can easily be verified that the inverse matrices are related by the equation

$$A_{X'}^{-1} = E^r A_X^{-1}, \tag{5.7}$$

where

$$E^r = \left\| \begin{matrix} 1 & 0 & \ldots & y_{1k} & \ldots & 0 \\ 0 & 1 & \ldots & y_{2k} & \ldots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \ldots & y_{rk} & \ldots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \ldots & y_{mk} & \ldots & 1 \end{matrix} \right\|,$$

$$y_{ik} = -\frac{x_{ik}}{x_{rk}}, \quad i = 1, 2, \ldots, m, \quad i \neq k; \quad y_{rk} = \frac{1}{x_{rk}}.$$

Relationship (5.7) is equivalent to the application of recurrence formulas (1.21) for $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, m$.

Generally, the solution of the linear-programming problem starts with the unit basis. This corresponds to the unit matrix E. Therefore, after the first iteration, when the basis vector $A_{s_{r_1}}$ has been replaced by the vector A_{k_1}, the inverse of the basis matrix of the new support program X' can be determined from

$$A_{X'}^{-1} = E^{r_1} E,$$

and, after the l-th iteration, from

$$A_{X^{(l)}}^{-1} = E^{r_l} E^{r_{l-1}} \ldots E^{r_1} E. \tag{5.8}$$

The matrix E^r is specified by the $m+1$ number $(r, y_{1k}, \ldots, y_{mk})$. Hence, for $l < m$ keeping the inverse matrix in the form (5.8) less of the memory is occupied than in the ordinary procedure, where $m \times m$ numbers $e_{ij}^{(l)}$ must be stored in each iteration.

As already indicated, from experience we know that the number of iterations required for solving the linear-programming problem is, generally, of the order m. It should, therefore, be expected that the product form will almost always prove more compact than the original form of the second algorithm. In those cases when the number of iterations is greater than m and for some reason it is inexpedient to transform the inverse of the basis matrix of the support program X^m to the unit matrix*, we should, starting with the $(m+1)$-iteration, revert to the original procedure of storing the inverse.

Let us now outline the computational procedure for a single iteration in the product form of the second algorithm.

Assume that the l-th iteration has been completed and it has been established that $X^{(l)}$ is not an optimal program but there is no reason to suspect that the problem is unsolvable. The vector to be introduced into the basis and that to be eliminated should, obviously, be determined by the general simplex method. The components of the current support program are computed, as in the original form of the second algorithm, from recurrence formulas (1.21).

The evaluation vector of problem restraints (Λ) is computed in the $(l+1)$-th iteration as follows

$$\Lambda_{l+1} = C_X E^{r_l} E^{r_{l-1}} \ldots E^{r_1} E.$$

The evaluation vector of the restraint vectors is determined as always under the second algorithm from

$$\Delta = \Lambda A_X - C.$$

Finally the coefficients x_{ik} in the expansion of the vector A_k to be introduced into the basis in terms of the basis vectors are determined from

$$A_k^{(l+1)} = E^{r_l} E^{r_{l-1}} \ldots E^{r_1} A_k.$$

Here

$$A_k^{(l+1)} = (x_{1k}^{(l+1)}, \ldots, x_{mk}^{(l+1)}); \quad A_k = A_k^{(0)} = (a_{1k}, \ldots, a_{mk})^T.$$

We see that the product form of the second algorithm not only economizes on memory storage space, but also in some cases essentially reduces, at least in the first iterations, the bulkiness of computations.

5-5. Let us consider the specific features of the second algorithm in connection with the solution of degenerate problems. We have already indicated that unless cycling has been detected it is advisable to employ the simplified rule for determining which vector is to be eliminated from the basis. When a cycle arises, the index r should be computed by the rule given in Chapter 4, § 6. The unit vectors e_j suggest themselves as the vectors R_i, since in each iteration we compute the coefficients of e_j when

* Transformation of the inverse A_X^{-1} to the unit matrix in any single iteration corresponds to transition to the equivalent problem whose restraint vectors and constraint vector are obtained from the restraint vectors and the constraint vector of the initial problem when these are multiplied by A_X^{-1} on the left.

expanded in terms of the basis vectors. This system of vectors is linearly independent. The vectors e_j, however, need not satisfy the second requirement to be filled by the system R_i. The coefficients in the expansion of the vector $e_0 + \sum_{j=1}^{m} \varepsilon^j e_j$ in terms of the basis vectors may prove to be negative. In the principal part of some tableaus a row whose first nonzero entry is a negative coefficient $e_{ij} (0 \leqslant i \leqslant m)$ may appear. To avoid this, it is advisable, once cycling has occurred, to introduce into the principal tableaus an additional column W comprising only positive numbers, e. g., $w_i = 1, i = 1, 2, \ldots, m$, and to take $R_1 = A_X W$, $R_2 = e_1, \ldots, R_t = e_{t-1}$, $R_{t+1} = e_{t+1}, \ldots$, $R_m = e_m$. Here A_X is the basis matrix; the index t is chosen so that the t-th coefficient of the vector $A_X W$ is nonzero. The resulting system of vectors, obviously, satisfies all the conditions imposed on R_i. It must be remembered that when passing from tableau to tableau the column W should also be transformed according to the general rules.

The additional column becomes unnecessary if the rule for avoiding cycling is applied at the outset under the second algorithm. The vectors e_j $(j = 1, \ldots, m)$ may be used as the system R_i $(i = 1, 2, \ldots, m)$ also if, when cycling is detected, the inverse matrix is transformed to the unit matrix. It seems, however, that transformation of the additional column W in several successive iterations will involve less computational work on the whole than application of the complex degeneracy rule throughout the solution process. Transformation to the unit basis is also rather tedious.

The position of the vector to be eliminated from the basis is determined as follows. Let $\theta_0 = \min\limits_i \frac{e_{i0}}{x_{ik}}$ be obtained on several positions. We compute the minimum of the ratios $\theta_i' = \frac{u_i}{x_{ik}}$ for the indices i on which θ_0 is obtained (or $\min \frac{e_{i1}}{x_{ik}}$, if column W is not required). If here θ_0 is not unique, we compute the ratios $\theta_i'' = \frac{e_{i1}}{x_{ik}}$ (or $\theta_i'' = \frac{e_{i2}}{x_{ik}}$, if column W is not introduced), etc., until the minimum of the ratio is obtained on a single vector. This vector is then eliminated from the basis.

§ 6. Examples and comparative evaluation of the algorithms

6-1. We shall now illustrate the computational procedure of the second simplex algorithm by the two examples discussed in § 3 in connection with the first algorithm.

In the principal tableaus, Tables 5.6, and in the auxiliary tableau, Table 5.7, we give the process of solution of Example 1 (see § 3). Tables 5.6 are divided into 8 tableaus. The principal part of the zeroeth tableau contains the initial data for the first iteration. The last two columns and the last row of the principal and the auxiliary tableaus, together with the principal part of the subsequent principal tableau and the Δ-th row of the auxiliary tableau correspond to the current iteration of the algorithm. In the upper part of the auxiliary tableau the components of the constraint vector and of all the restraint vectors are written. In the $(m+1)$-th (sixth) row the corresponding coefficients of the linear form are written. The $\Delta^{(l)}$ rows in each iteration are filled from formulas (5.1). The initial basis comprises the unit vectors corresponding to the additional variables $x_7, x_8, \ldots, x_{11}$. The corresponding linear-form coefficients are all zero. Therefore, in the principal zeroeth tableau all $\lambda_j = 0$ and in the row 0 of the auxiliary tableau $\Delta_j = -c_j$. The vector A_6 with the least evaluation $(\Delta_6 = -9)$ is to be introduced into the basis.

In the column A_k of the principal zeroeth tableau the coefficients x_{i6} in the expansion of A_6 in terms of the basis vectors are written. Since the initial basis consists of unit vectors, $x_{i6} = a_{i6}$. In the $(m+1)$-th cell of the column A_k we write the evaluation Δ_6 of A_6. Column A_3 contains positive components. We thus have case (c).

TABLE 5.6 (0–7)

	No.	c_X	B_X	e_0	e_1	e_2	e_3	e_4	e_5	A_k	θ	No. of tableau
	1		A_7	12	1					—1	—	
	2		A_8	5		1				—1	—	
	3		A_9	20			1			4	5	0
←	4		A_{10}	10				1		4	2.5	
	5		A_{11}	24					1	5	4.8	
	6	—	—							—9	—	
	1		A_7	14.5	1			0.25		—2	—	
	2		A_8	7.5		1		0.25		—5	—	
←	3		A_9	10.0			1	—1 00		14	0.714	1
→	4	9	A_6	2.5				0.25		—2	—	
	5		A_{11}	11.5				—1.25	1	14	0.822	
	6	—	—	22.5				2.25		—20	—	
	1		A_7	15.929	1		0.143	0.107		1.857	8.577	
	2		A_8	11.071		1	0.357	—0.107		1.643	6.799	
→	3	2	A_4	0.714			0.071	—0.71		—0.071	—	2
	4	9	A_6	3,929			0.143	0 107		—1.143	—	
←	5		A_{11}	1.500			—1.000	—0.250	1	8.000	0.188	
	6	—	—	36.786			1.429	0.821		—18.429	—	

TABLE 5.6 (continued)

	No	c_X	B_X	e_0	e_1	e_2	e_3	e_4	e_5	A_k	θ	No. of tableau
	1		A_7	15.580	1		0.375	0.165	−0.232	11.272	1.382	3
	2		A_8	10.763		1	0.562	−0.056	−0.205	0.462	23.295	
←	3	2	A_4	0.728			0.062	−0.074	0.009	0 730	0.997	
	4	9	A_6	4.143				0.071	0 143	−0.071	—	
→	5	8	A_3	0.188			−0.125	−0.031	0.125	−0.781	—	
	6	—	—	40.241			−0.875	0.246	2.304	−4.333	—	
	1		A_7	4.343	1		−0.590	1.303	−0.370	−0.590	—	4
	2		A_8	10.303		1	0.523	−0.009	−0.211	0.523	19.702	
⇆	3	−1	A_2	0.997			0.086	−0.101	0.012	0.086	11.643	
	4	9	A_6	4.214			0.006	0.064	0.144	0.006	689.0	
	5	8	A_3	0.966			−0.058	−0.110	0.135	−0.058	—	
	6	—	—	44.661			−0.495	−0.202	2.358	−0.495	—	
	1		A_7	11.214	1			0.607	−0.286	0.607	18.471	5
←	2		A_8	4.214		1		0.607	−0.286	0.607	6.941	
→	3		A_9	11.643			1	−1.179	0.143	−1.179	—	
	4	9	A_6	4.143				0.071	0.143	0.071	58.00	
	5	8	A_3	1.643				−0.179	0.143	−0.179	—	
	6	—	—	50.429				−0.786	2.429	−0.786	—	

TABLE 5.6 (continued)

	No.	c_X	B_X	e_0	e_1	e_2	e_3	e_4	e_5	A_k	θ	No. of tableau
←	1		A_7	7.000	1	−1.000				13.000	0.538	6
→	2		A_{10}	6.941		1.647		1	−0.471	−10.059	—	
	3		A_9	19.824		1.941	1		−0.412	−0.176	—	
	4	9	A_6	3.647		−0.118			0.176	0.647	5.636	
	5	8	A_3	2.882		0.294			0.059	−1.118	—	
	6	—	—	55.882		1.294			2.059	−2.118	—	
→	1	−1	A_2	0.538	0.077	−0.077						7
	2		A_{10}	12.357	0.774	0.873		1	−0.471			
	3		A_9	19.919	0.014	1.928	1		−0.412			
	4	9	A_6	3.299	−0.050	−0.068			0.176			
	5	8	A_3	3.484	0.086	0.208			0.059			
	6	—	—	57.023	0.163	1.131			2.059		—	

TABLE 5.7

No.	B	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}
1	12	−6	9	3		−2	−1	1				
2	5		−4	3	−3	1	−1		1			
3	20	2	8	−5	6	−8	4			1		
4	10	−1	−3	−4	−8		4				1	
5	24	5	1	2	4	9	5					1
6	C	3	−1	8	2	−1	9					
0	Δ	−3	1	−8	−2	1	**−9**					
1	Δ	−5.25	−5.75	−17	**−20**	1					2.25	
2	Δ''	−0.964	9.964	**−18.429**		−10.429				1.429	0.821	
3	Δ'''	6.522	**−4.433**			28.732				−0.875	0.246	2.304
4	Δ^{IV}	8.000			6.073	26.183				**−0.495**	−0.202	2.358
5	Δ^{V}	9.929	5.786		14.000	22.857					**−0.786**	2.429
6	Δ^{VI}	7.294	**−2.118**		2.353	20.824			1.294			2.059
7	Δ^{VII}	0.632			2.842	20.335		0.163	1.131			2.059

TABLE 5.8 (0–5)

	No.	C_X	B_X	e_0	e_1	e_2	e_3	e_4	e_5	e_6	A_k	θ	No. of tableau
←	1		A_{10}	24	1						6	4	0
	2		A_{11}	30		1						—	
	3		A_{12}	40			1				9	4.444	
	4		A_{13}	36				1				—	
	5		A_{14}	20					1			—	
	6		A_{15}	48						1	3	16	
	7	—	—								—9	—	
→	1	9	A_1	4	0.167							—	1
	2		A_{11}	30		1						—	
	3		A_{12}	4	—1.5		1					—	
	4		A_{13}	36				1			1	36	
	5		A_{14}	20					1			—	
←	6		A_{15}	36	—0 5					1	4	9	
	7	—	—	36	1.5						—8	—	
	1	9	A_1	4	0.167						0.5	8	2
	2		A_{11}	30		1					9	3.333	
	3		A_{12}	4	—1.5		1				—4.5	—	
	4		A_{13}	27	0.125			1		—0.25	0.375	72	
←	5		A_{14}	20					1		8	2.5	
→	6	8	A_9	9	—0.125					0.25	—0.375		
	7	—	—	108	0.5					2	—6.5	–	

TABLE 5.8 (continued)

	No.	c_X	B_X	e_0	e_1	e_2	e_3	e_4	e_5	e_6	A_k	θ	No. of tableau
	1	9	A_1	2.75	0.167				−0.062		0 333	8.25	3
	2		A_{11}	7.5		1			−1.125			—	
←	3		A_{12}	15.25	−1.5		1		0.562		2	7.625	
	4		A_{13}	26.062	0.125			1	−0.047	−0.25	0.25	104.25	
→	5	8	A_8	2.5					0.125			—	
	6	8	A_9	9.938	−0.125				0.047	0.25	−0.25	—	
	7	—	—	124.25	0.5				0.812	2	−4	—	
←	1	9	A_1	0.208	0.417		−0 167		−0.156		0.417	0.5	4
	2		A_{11}	7.5		1			−1.125			—	
→	3	5	A_5	7.625	−0.75		0.5		0.281		−0.75	—	
	4		A_{13}	24.156	0.312		−0.125	1	−0.117	−0.25	0 312	77.3	
	5	8	A_8	2.5					0.125			—	
	6	8	A_9	11.844	−0.312		0.125		0.117	0.25	−0.312	—	
	7	—	—	154.75	−2 5		2		1.938	2	−2.5	—	
→	1		A_{10}	0.5	1		−0.4		−0.375				5
	2		A_{11}	7.5		1			−1.125				
	3	5	A_5	8			0.2						
	4		A_{13}	24				1		−0.25			
	5	8	A_8	2.5					0.125				
	6	8	A_9	12						0.25			
	7	—	—	156			1		1	2		—	

TABLE 5.9

No.	B	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
1	24	6		2		2			3		1					
2	30		2		3			5	9			1				
3	40	9	1			5	1						1			
4	36			6				2		1				1		
5	20				4				8						1	
6	48	3		1			8	4		4						1
7	C	9		—8		5	—3		8	8						
0	Δ	**—9**		8		—5	3		—8	—8						
1	Δ'			11		—2	3		—3.5	—8	1.5					
2	Δ''			11		—4	19	8	—6.5		0.5					
3	Δ'''			11	3.25	**—4**	19	8			0.5				—0.812	2
4	Δ^{IV}		2	5	7.75		21	8			**—2.5**		2		1.938	2
5	Δ^{V}	6	1	10	4		20	8					1		1	2

TABLE 5.10 (0–6)

	No.	C_X	B_X	e_0	e_1	e_2	e_3	e_4	e_5	A_k	θ	No. of tableau
	1		A_7	12	1					−6	—	0
	2		A_8	5		1					—	
	3		A_9	20			1			2	10	
	4		A_{10}	10				1		−1	—	
←	5		A_{11}	24					1	5	**4.8**	
	6	—	—							3	—	
	1		A_7	40.8	1				1.2	5.4	7.556	1
←	2		A_8	5		1				3	**1.667**	
	3		A_9	10.4			1		−0.4	−5.8	—	
	4		A_{10}	14.8				1	0.2	−3.6	—	
→	5	3	A_1	4.8					0.2	0.4	12	
	6	—	—	14.4					0.6	−6.8	—	

TABLE 5.10 (continued)

	No.	c_X	B_X	e_0	e_1	e_2	e_3	e_4	e_5	A_k	θ	No. of tableau
←	1		A_7	31.8	1	−1.8			1.2	10.2	**3.118**	2
→	2	8	A_3	1.667		0.333				−1	—	
	3		A_9	20.067		1.933	1		−0.4	−1.4	—	
	4		A_{10}	20.8		1.2		1	0.2	−10.8	—	
	5	3	A_1	4.133		−0.133			0.2	1.2	3.444	
	6	—	—	25.733		2.267			0.6	−6.4	—	
→	1	2	A_4	3.118	0.098	−0.176			0.118	0.667	4.676	3
	2	8	A_3	4.784	0.098	0.157			0.118	0.333	14.353	
	3		A_9	24.431	0.137	1.686	1		−0.235	1	24.431	
	4		A_{10}	54.471	1.059	−0.706		1	1.471	11	4.952	
←	5	3	A_1	0.392	−0.118	0.078			0.059	0.333	**1.176**	
	6	—	—	45.686	0.627	1.137			1.353	−4	—	
←	1	2	A_4	2.333	0.333	−0.333				0.333	7	4
	2	8	A_3	4.392	0.216	0.078			0.059	0.216	20.364	
	3		A_9	23.255	0.490	1.451	1		−0.412	0.490	47 440	
	4		A_{10}	41.529	4.941	−3.294		1	−0.471	4.941	8.405	
→	5	9	A_6	1.176	−0.353	0.235			0.176	−0.353	—	
	6	—	—	50.392	−0.784	2.078			2.059	−0.784	—	

TABLE 5.10 (continued)

	No.	c_X	B_X	e_0	e_1	e_2	e_3	e_4	e_5	A_k	θ	No. of tableau
⇄	1		A_7	7	1	—1				13	0.538	5
	2	8	A_3	2.882		0.294			0.059	—1.118	—	
	3		A_9	19.824		1.941	1		—0.412	—0.176	—	
	4		A_{10}	6.941		1.647		1	—0.471	—10.059	—	
	5	9	A_6	3.647		—0.118			0.176	0.647	5,636	
	6	—	—	55.882		1.294			2.059	—2.118	—	
→	1	—1	A_2	0.538	0.077	—0.077						6
	2	8	A_3	3.484	0.086	0.208			0.059			
	3		A_9	19.919	0.014	1.928	1		—0.412			
	4		A_{10}	12.357	0.774	0.873		1	—0.471			
	5	9	A_6	3.299	—0.050	—0.068			0.176			
	6	—	—	57.023	0.163	1.131			2.059			

TABLE 5.11

No.	B	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}
1	12	−6	9	3		−2	−1	1				
2	5		−4	3	−3	1	−1		1			
3	20	2	8	−5	6	−8	4			1		
4	10	−1	−3	−4	−8		4				1	
5	24	5	1	2	4	9	5					1
6	C	3	−1	8	2	−1	9					
0	Δ	**−3**										
1	Δ'		1.6	**−6.8**								
2	Δ''				**−6.4**							
3	Δ'''					13.059	**−4**					
4	Δ^{IV}							**−0.784**				
5	Δ^{V}	7.294	**−2.118**						1.294			2.059
6	Δ^{VI}	6.317			2.842	20.335		0.163	1.131			2.059

We now proceed to determine the vector to be eliminated from the basis. In column θ we fill the cells for which $x_{ik} > 0$ (cells 3, 4, and 5). The least value $\theta = \theta_0 = 2.5$ corresponds to the basis vector A_{10} occupying the fourth position in the basis. The fourth row is, thus, the direction row of the transformation, and the vector A_{10} is to be replaced by A_6. The direction element of the transformation is $x_{46} = 4$, occupying the intersection of the fourth row and column A_6.

The principal part of the 1-th tableau is computed from the elements of the principal part of the zeroeth tableau with the aid of the recurrence formulas. The entries of the fourth row in the 1-th tableau are equal to the corresponding entries in the fourth row of the zeroeth tableau divided by the direction element $x_{46} = 4$. All the remaining elements of the principal part of the tableau (including the components of the column e_0 and the row λ) are computed from formula (5.6). For example,

$$x'_{30} = e'_{30} = e_{30} - \frac{e_{40}}{x_{46}} x_{36} = 20 - \frac{10 \cdot 4}{4} = 10,$$

$$e'_{54} = e_{54} - \frac{e_{44}}{x_{46}} x_{56} = 0 - \frac{1 \cdot 5}{4} = -1.25,$$

$$L(X') = e'_{m+1,0} = \lambda'_0 = e'_{60} = e_{60} - \frac{e_{40}}{x_{46}} x_{66} = 0 - \frac{10(-9)}{4} = 22.5.$$

Here $x_{66} = \Delta_6$ is the evaluation of the vector A_6 introduced into the basis.

Now, using (5.1), we compute the first row (Δ') of the auxiliary tableau, Table 5.7. We have, for instance,

$$\Delta'_4 = (0 \cdot 0 - 3 \cdot 0 + 6 \cdot 0 - 8 \cdot 2.25 + 4 \cdot 0) - 2 = -20.$$

Δ'_4 is the least element in the Δ' row. A_4 is thus introduced into the current basis. Subsequent iterations follow the same rules. For control purposes it is advisable in some iterations to compute the restraint evaluations λ_i not only from recurrence formulas (5.5), but also directly from (5.3).

In Tables 5.8 and 5.9 we present the sequence of computations leading to the determination of the optimal program of Example 2 (see § 3) according to the second algorithm. The tables do not require any special explanation.

In Tables 5.10 and 5.11 we give a solution of Example 1 (see Tables 5.2) following the procedure in which the vector to be introduced into the basis in each iteration is computed according to the rule given in 5-3. We see that the number of iterations in this modification of the second algorithm is less by one. The reduction in the number of iterations, generally speaking, is not characteristic of this rule for selecting vectors to be introduced into the basis. Comparing the auxiliary tableaus in Tables 5.7 and 5.11 we see that the computations are less bulky. In Table 5.11 fewer evaluations Δ_j had to be computed.

6-2. We now compare the two simplex algorithms. Computations under the first algorithm involve only one kind of tableaus and are standardized to a higher degree than the computations under the second algorithm. In manual calculations this is a considerable advantage. Moreover, the first algorithm will establish unsolvability of the problem sooner than the second algorithm. As we shall see in § 8, the first algorithm is more convenient for solving the auxiliary problem in which the initial support program of the principal problem is determined. Finally, the first algorithm ensures that the vector A_k introduced into the basis will produce the maximum possible increase of the linear form in that iteration:

$$-\Delta_k \theta_0^{(k)} = \max_j (-\Delta_j \theta_0^{(j)}).$$

Computations under the second algorithm require less memory storage space than computations under the first algorithm. Under the first algorithm, the array corresponding to all the restraint vectors is stored. The volume of information stored under the second algorithm is determined only by the basis vectors. In the product form of the second algorithm the volume of data stored in each iteration is still less.

In applied problems the percentage of zeros in the restraint matrix is generally very high. In the first algorithm, after elementary transformation the zero components of the restraint vectors are converted, generally speaking, into nonzero components. Under the second algorithm, in which

the restraint matrix as a whole is not transformed from table to table, the relative advantage of the restraint vectors with numerous zero components is retained throughout the computations. In particular, the unit restraint vectors are very easily evaluated under the second algorithm:

$$\Delta_j = \sum_{i=1}^{m} a_{ij}\lambda_i - c_j = \sum_{i=1}^{m} \delta_i^j \lambda_i - c_j = \lambda_j - c_j.$$

When solving a problem following the second algorithm, the optimal program of the primal problem is obtained simultaneously with the optimal program of the dual problem (the vector $\Lambda^* = (\lambda_1^*, \ldots, \lambda_m^*)$, occupying the $(m+1)$-th row of the last principal tableau) and with the inverse matrix of the optimal basis. This feature of the second algorithm is most useful in some applications.

When solving a problem according to the second algorithm, we refer more often to the initial data than with the first algorithm. It is, therefore, obvious that computational errors in the second algorithm accumulate more slowly than under the first.

Let us now compare the two algorithms with regards to the number of multiplications and divisions per iteration.

In 2-4 we showed that the solution of the linear-programming problem according to the first algorithm requires at most $n+1$ divisions and $m(n-m+1)$ multiplications in each iteration. In 5-3 we established that each iteration in the second algorithm involves at most $2m+1$ divisions and $m(n+m+1)$ multiplications. We see that the second algorithm, involving transformation of the inverse matrix and determination of the vector to be introduced into the basis from the least evaluation Δ_j, involves, generally speaking, more operations than the first algorithm.

The number of multiplications in each iteration of the second algorithm can be reduced if, as indicated in 5-3, the vector to be introduced in the basis is chosen independently of the computation of the Δ_j of all the restraint vectors. In this case, however, the computational bulkiness of the two algorithms cannot be compared with respect to the number of operations per iteration, since the solution of a problem according to different algorithms will involve, generally speaking, a different number of iterations.

In some cases computational bulkiness is somewhat reduced by the product form of the second algorithm.

Observe that the comparative evaluation of the computational bulkiness of the first and the second algorithms in Wagner's paper /13/, which is quoted in certain works, cannot be considered objective. In this paper it is indicated that with $n > 3m$ the first algorithm invariably involves bulkier computations than the second algorithm. When determining the number of multiplications performed under the first algorithm, Wagner assumes that the evaluations Δ_j are computed in the first algorithm not from the recurrence formulas (2.2), but directly from (5.1). Wagner's conclusion would apply if it were found necessary to check computations after each iteration. In Table 5.12 we give the number of multiplications required for a single iteration and for control purposes in the first and the second simplex algorithms. We see from the table that for $n > 3m$ the first algorithm is bulkier than the second, It is, however, quite unnecessary to check the computations in each step of the method.

This comparison of the two algorithms does not take into consideration the specific features of particular classes of linear-programming problems.

In problems with a high percentage of zeros in the restraint matrix (these problems are quite frequent in applications), the number of operations per iteration under the second algorithm is less than under the first. We have already seen that the most tedious stage of computations under the second algorithm – determination of the evaluations Δ_j – is essentially reduced for restraint vectors containing numerous zeros.

TABLE 5.12

	1st algorithm	2nd algorithm
Per iteration	$m(n-m+1)$	$m(n+m+1)$
Check	$m(n+1)$	$m(m+1)$
Total	$m(2n-m+2)$	$m(2m+n+2)$

Let each column of the restraint matrix contain at most αm nonzero elements. The computation of Δ_j involves $\alpha m(n-m)$ multiplications. The transformation of the principal part of the tableau involves, as before, $m(m+1)$ multiplications, and the elements of the direction column are computed in $A_k^{(l)}-m^2$ multiplications. The total number of multiplications in each iteration in the second algorithm are, thus, at most

$$\alpha m(n-m)+m(m+1)+m^2=m[\alpha n+(2-\alpha)m+1].$$

The bulkiness of computations under the first algorithm is independent of the number of zeros in the initial restraint matrix. The total number of multiplications in each iteration in the first algorithm is $m(n-m+1)$.

Thus, when

$$m[\alpha n+(2-\alpha)m+1]<m(n-m+1),$$

i.e., when

$$n>\frac{3-\alpha}{1-\alpha}m,$$

the second algorithm involves less multiplications than the first.

Analogous evaluations can be made for other particular classes of linear-programming problems.

The comparison of the two algorithms discussed in this section shows that one algorithm may be preferable to the other depending on the problem under consideration.

§ 7. The case of bilateral restraints

7-1. In Chapter 4, § 5, we showed that when linear-programming problems are subject to bilateral restraints, the simplex computational procedure remains essentially the same and only some elements of each iteration become more complicated. The modifications of the algorithm pertain, mainly, to the method of determining which vector is to be

eliminated from the basis, the transformation formulas of the basis variables (the coefficients in the expansion of the vector

$$A_0 = B - \sum_{j \notin I_X} x_j A_j),$$

and the evaluations Δ_j of the restraint vectors.

Consider the modified structure of the tableaus for a problem with bilateral restraints solved by the first algorithm. The l-th tableau corresponding to the l-th iteration of the problem with bilaterally restrained variables differs from the l-th tableau compiled in the solution process of the problem in canonical form in that there are three additional rows and two additional columns (see Table 5.13). In the row (α, β) of this table, above each restraint vector not included in the basis, we write the value of the corresponding extrabasis variable. If x_j is equal to its lowest boundary value, this should be signified by the letter α, and if x_j is equal to the highest boundary value, this should be signified by the letter β. The entries in the (α, β) row corresponding to the basis vectors are crossed out.

TABLE 5.13
l-th tableau

			(α, β)									
No.	c_X	B_X	A_0	A_1	A_2	...	A_k	...	A_n	$\tilde{A}_k$	γ	θ
1	c_{s_1}	A_{s_1}	$x_{10}^{(l)}$	$x_{11}^{(l)}$	$x_{12}^{(l)}$	...	$x_{1k}^{(l)}$	...	$x_{1n}^{(l)}$	$\tilde{x}_{1k}^{(l)}$	$\gamma_1^{(l)}$	
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	...	⋮	⋮	⋮	⋮
r	c_{s_r}	A_{s_r}	$x_{r0}^{(l)}$	$x_{r1}^{(l)}$	$x_{r2}^{(l)}$	...	$x_{rk}^{(l)}$	...	$x_{rn}^{(l)}$	$\tilde{x}_{rk}^{(l)}$	$\gamma_r^{(l)}$	$\theta_0^{(l)}$
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	...	⋮	⋮	⋮	⋮
m	c_{s_m}	A_{s_m}	$x_{m0}^{(l)}$	$x_{m1}^{(l)}$	$x_{m2}^{(l)}$	...	$x_{mk}^{(l)}$	...	$x_{mn}^{(l)}$	$\tilde{x}_{mk}^{(l)}$	$\gamma_m^{(l)}$	
$m+1$	—	A_k	$x_k^{(l)}$	—	—	...	-1	...	—	$(-1)^{\gamma+1}$	$\gamma_{m+1}^{(l)}$	$\theta_{m+1}^{(l)}$
$m+2$	—	—	$x_{m+2,0}^{(l)}$	$x_{m+2,1}^{(l)}$	$x_{m+2,2}^{(l)}$	...	$x_{m+2,k}^{(l)}$	...	$x_{m+2,k}^{(l)}$	$\Delta_k^{(l)}$	—	—
$m+3$	—	—	$L^{(l)}$	$\Delta_1^{(l)}$	$\Delta_2^{(l)}$	...	$\Delta_k^{(l)}$	...	$\Delta_n^{(l)}$	—	—	—

The evaluations $\Delta_j^{(l)}$, which in the case of unilateral restraints (see § 3) were written in the $(m+1)$-th row, are written in the case of bilateral restraints in the $(m+3)$-th row of the tableau. In this case

$$\Delta_j^{(l)} = x_{m+3,\,j}^{(l)}; \quad x_{m+3,\,0}^{(l)} = L(X^{(l)}).$$

The elements written in the $(m+1)$-th and $(m+2)$-th rows will be specified below.

To the right of the columns corresponding to the restraint vectors we have two new columns, $\tilde{A}_k$ and γ. The column $\tilde{A}_k$ contains the coefficients in the expansion of the vector A_k in terms of the basis vectors, written

with their respective sign if $x_k = \alpha_k$ and with reverse signs if $x_k = \beta_k$ (A_k is the vector to be introduced into the basis). In other words, the elements of the column $\tilde{A}_k$ are equal to

$$\tilde{x}_{ik}^{(l)} = (-1)^{\gamma} x_{ik}^{(l)},$$

where $\gamma = 0$ for $x_k = \alpha_k$ and $\gamma = 1$ for $x_k = \beta_k$.

In the next column, γ, we write α_i if the corresponding position of the column $\tilde{A}_k$ is occupied by a positive number, and β_i if $\tilde{x}_{ik}^{(l)} < 0$, i.e.,

$$\gamma_i^{(l)} = \begin{cases} \alpha_i, & \text{if } \tilde{x}_{ik}^{(l)} > 0, \\ \beta_i, & \text{if } \tilde{x}_{ik}^{(l)} < 0. \end{cases} \tag{7.1}$$

The entries in the column γ for which $\gamma_i^{(l)} = -\infty$ or $\gamma_i^{(l)} = \infty$ and likewise the positions corresponding to $x_{ik}^{(l)} = 0$ are crossed out.

Each entry in the last column θ is computed as the difference of the corresponding elements in columns A_0 and γ divided by the element of the column $\tilde{A}_k$ in the same row:

$$\theta_i^{(l)} = \frac{x_{i0}^{(l)} - \gamma_i^{(l)}}{\tilde{x}_{ik}^{(l)}}. \tag{7.2}$$

In the $(m+1)$-th row of the table only the five positions corresponding to the columns A_0, A_k, $\tilde{A}_k$, γ, and θ are filled. The $(m+1)$-th entry in the column A_0 is the parameter $x_{m+1,0}^{(l)} = x_k^{(l)}$ (α_k or β_k), i.e., the value of the extrabasis variable corresponding to the vector A_k introduced into the basis. The $(m+1)$-th entry in column A_k is always the number $x_{m+1,k}^{(l)} = -1$. The values of $\tilde{x}_{m+1,k}^{(l)}$, γ_{m+1} and θ_{m+1} are computed according to the general rules for computing the entries of the columns $\tilde{A}_k^{(l)}$, γ, and θ:

$$\tilde{x}_{m+1,k}^{(l)} = x_{m+1,k}^{(l)}(-1)^{\gamma} = (-1)^{\gamma+1},$$

$$\gamma_{m+1}^{(\alpha)} = \begin{cases} \alpha_k, & \text{if } x_k^{(l)} = \beta_k, \\ \beta_k, & \text{if } x_k^{(l)} = \alpha_k, \end{cases}$$

$$\theta_{m+1} = \frac{x_{m+1,0}^{(l)} - \gamma_{m+1}^{(l)}}{\tilde{x}_{m+1,k}^{(l)}} = \beta_k - \alpha_k.$$

In the $(m+2)$-th row we write the parameters $x_{m+2,j}^{(l)}$ $(j = 0, 1, \ldots, m)$ required for computing the evaluations $\Delta_j^{(l)}$. Formulas (7.4) for the computation of $x_{m+2,j}^{(l)}$ are given in 7-2.

In the case of problems with bilateral restraints, the principal part of the l-th tableau in the first algorithm comprises all the entries of columns $A_0, A_1, \ldots, A_n$ with the exception of the $(m+1)$-th and the $(m+3)$-th rows.

As in the general case, the form of the initial zeroeth tableau differs from the form of the l-th tableau in the additional row C containing the linear-form coefficients c_j. The entries of the zeroeth tableau are not computed from recurrence formulas. The parameters x_{ij} are computed from systems of equations (see (5.5) in Chapter 4), and the evaluations Δ_j are obtained from (5.6) and (5.7) in Chapter 4. Using any of the methods of determining the initial support program it is possible to obtain at the same time the corresponding values of the coefficients x_{ij} and the evaluations Δ_j.

7-2. We now outline the sequence of computations to be followed in the solution of a linear-programming problem with bilateral restraints according to the first simplex algorithm.

Assume that the l-th iteration has been completed. Thus, the principal

part of the l-th tableau and the $(m+3)$-th row are filled. The computations in the $(l+1)$-th iteration will fill the $(m+1)$-th row, the columns $\tilde{A}_k$, γ, and θ of the l-th tableau, and the principal part of the $(l+1)$-th tableau. In the first stage of the $(l+1)$-th iteration the $(m+3)$-th row of the l-th tableau is examined. If all the $\Delta_j^{(l)} \geqslant 0$, case (a) applies and the support program obtained in the l-th iteration solves the problem. If there are vectors with negative evaluations $(\Delta_j < 0)$, case (b) or case (c) may apply. Case (b) applies if, for some j,

(i) when $x_j = \alpha_j$, for $x_{ij} > 0$ $\alpha_{s_i} = -\infty$, and for $x_{ij} < 0$ $\beta_{s_i} = \infty$, or

(ii) when $x_j = \beta_j$, for $x_{ij} > 0$ $\beta_{s_i} = \infty$, and for $x_{ij} < 0$ $\alpha_{s_i} = -\infty$. We shall refer to (i) and (ii) as the unsolvability criteria of a problem with bilateral restraints.

Case (b) indicates unboundedness of the linear form in the set of feasible programs of the problem.

If unsolvability has not been established (i. e., if case (c) applies), we must proceed with the second stage of the iteration. In the second stage, the following operations are carried out:

(1) the vector to be introduced into the basis is determined;

(2) the vector to be eliminated from the basis is determined;

(3) the coefficients $x_{ij}^{(l)}$ in the expansion of the restraint vectors in terms of the basis vectors $(i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$ are transformed;

(4) the basis variables $x_{i0}^{(l)}$ and the value $x_{m+1,0}^{(l)}$ of the linear form are transformed;

(5) the row (α, β) is transformed;

(6) the evaluations $\Delta_j^{(l)}$ of the restraint vectors are transformed.

We consider each of these operations separately.

(1) The vector A_k with the least evaluation $\Delta_k^{(l)}$ is to be introduced into the basis. Having chosen A_k, we fill the columns $\tilde{A}_k$, γ, and θ and the $(m+1)$-th row of the l-th tableau.

(2) The vector A_{s_r} for which

$$\theta_0^{(l)} = \min_i \theta_i^{(l)}$$

is to be eliminated from the basis. If the program is degenerate, $\theta_0^{(l)}$ is obtained on several vectors, the vector with the least index is eliminated. When cycling occurs, the more complicated rule described in 7-4 is applied.

(3) The coefficients $x_{ij}^{(l)}$ $(i = 1, \ldots, m;\ j = 1, \ldots, n)$ in the expansion of the restraint vectors in terms of the basis vectors are transformed according to the recurrence formulas (1.12) if $\theta_0^{(l)}$ is obtained on one of the first m positions of the basis. If $\theta_0^{(l)}$ is obtained on the $(m+1)$-th position of the basis, the basis remains as it was and the coefficients x_{ij} retain their original value, i. e., $x_{ij}^{(l+1)} = x_{ij}^{(l)}$ for $r = m+1$ $(i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$.

(4) The basis variables are transformed according to the recurrence formulas

$$x_{i0}^{(l+1)} = \begin{cases} x_{i0}^{(l)} - \theta_0^{(l)} \tilde{x}_{ik}^{(l)} \text{when } i \neq r, \\ x_{m+1,0} - \theta_0^{(l)} (-1)^{\nu+1} \text{when } i = r, \end{cases} \qquad (7.3)$$
$$1 \leqslant i \leqslant m.$$

Formulas (7.3) apply regardless of whether $\theta_0^{(l)}$ is obtained on the first m positions or on the $(m+1)$-th position of the basis. The same recurrence formulas, (7.3), are used to transform the value $x_{m+1,0}^{(l)}$ of the linear form. To this end the evaluation $\Delta_k^{(l)}$ of the vector to be introduced into the basis is written in the $(m+2)$-th cell of the column $(\tilde{x}_{m+2,k}^{(l)} = \Delta_k^{(l)})$.

(5) In the row (α, β), at most two entries are modified. For $r \leqslant m$ the vector A_k is introduced into the basis and, consequently, the entry in the row (α, β) corresponding to this vector is crossed out. In the position corresponding to the vector A_{s_r} eliminated from the basis we write the boundary value assumed by the component x_{s_r}:

$$x_{s_r}^{(l+1)} = x_{r0}^{(l)} - \theta_0^{(l)} \bar{x}_{rk}^{(l)} = \gamma_r^{(l)}.$$

For $r = m+1$ the basis remains unmodified. Therefore, the position corresponding to the vector A_{s_r} in the row (α, β) remains crossed out, and the boundary value of the component x_k in the row (α, β) is replaced by the opposite member of the pair.

(6) The relationship between $\Delta_j^{(l+1)}$ and $\Delta_j^{(l)}$ depends on whether the basis changes upon transition to the new program.

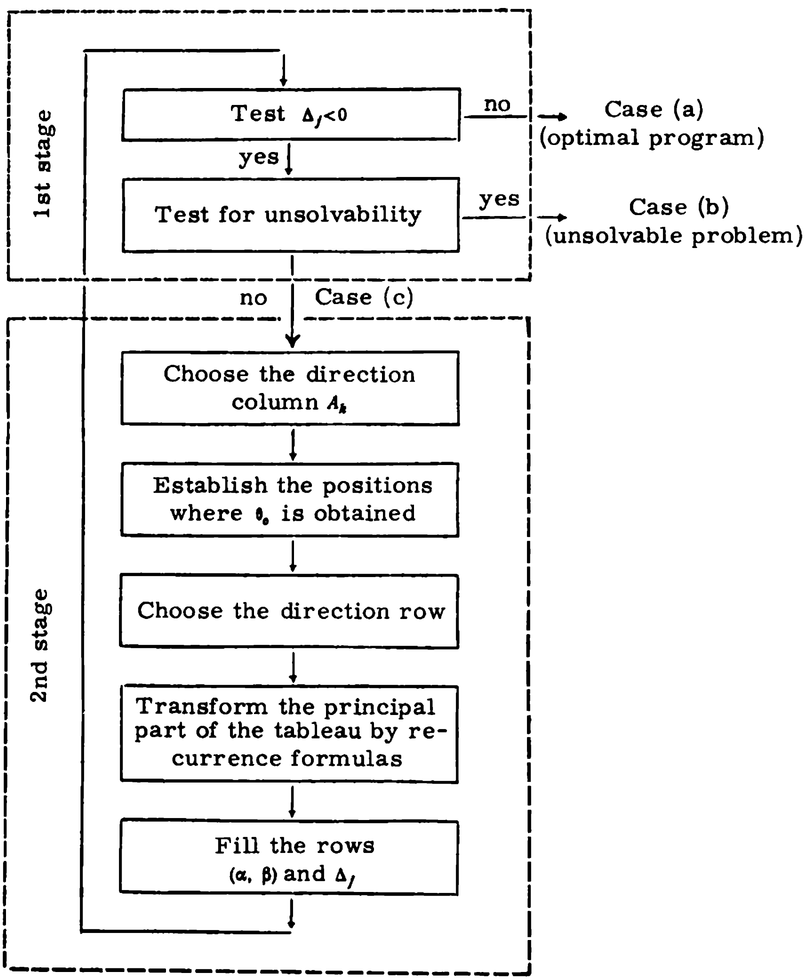


FIGURE 5.3

First let $r \leqslant m$. Further, let

$$x_{m+2,\,j}^{(l)} = z_j^{(l)} - c_j = \sum_{i=1}^{m} c_{s_i} x_{ij}^{(l)} - c_j, \quad j = 1, 2, \ldots, n. \tag{7.4}$$

Obviously,

$$\Delta_j^{(l)} = (-1)^{\nu_j} x_{m+2,\,j}^{(l)} = \tilde{x}_{m+2,\,j}^{(l)}, \quad j = 1, 2, \ldots, n, \tag{7.5}$$

where

$$\nu_j = 0, \quad \text{if} \quad x_j = \alpha_j, \text{ and } \nu_j = 1 \text{ for } x_j = \beta_j.$$

The parameters $x_{m+2,\,j}^{(l)}$ are written in the $(m+2)$-th row of the tableau and are transformed according to the same recurrence formulas (1.12) as the coefficients $x_{ij}^{(l)}$ in the expansion of the restraint vectors in terms of the basis vectors. Having transformed the parameters $x_{m+2,\,j}^{(l)}$ according to (7.5), we compute the transformed evaluations.

If $r = m+1$, the $(m+2)$-th row remains unmodified, except for the cell in which the value of the linear form is written. For $r = m+1$ the basis remains the same and the evaluations of all the restraint vectors, excepting A_k, are retained. The sign of the evaluation of the vector A_k is reversed. Thus, for $r = m+1$

$$\Delta_j^{(l+1)} = \begin{cases} \Delta_j^{(l)} & \text{for } j \neq k, \\ -\Delta_j^{(l)} & \text{for } j = k. \end{cases} \tag{7.6}$$

Having computed these parameters, the $(l+1)$-th iteration terminates and the principal part of the $(l+1)$-th tableau can be filled. Subsequent iterations follow the same rules.

In Figure 5.3 we give a block diagram of a single iteration in the solution of a linear-programming problem with bilateral restraints according to the first simplex algorithm.

7-3. We now outline the sequence of computations for the solution of a problem with bilateral restraints according to the second algorithm.

For the problem with bilateral restraints the structure of the principal and the auxiliary tableaus, which were compiled in § 5 for the linear-programming problem in canonical form, are slightly modified.

In the principal tableaus we add (as in the first algorithm) two columns, $\tilde{A}_k$ and γ. The columns are filled following the same rules as in the first algorithm.

The evaluations of problem restraints $\lambda_j^{(l)}$ $(j = 1, 2, \ldots, m)$ and $\lambda_0 = L(X)$ are written, not in the $(m+1)$-th row, as in the case of problems in canonical form, but rather in the $(m+2)$-th row. The $(m+1)$-th row is filled with the same parameters as the $(m+1)$-th row in the first algorithm. Five cells are filled in the $(m+1)$-th row:

$$e_{m+1,\,0}^{(l)} = x_k, \quad x_{m+1,\,k}^{(l)} = -1, \quad \tilde{x}_{m+1,\,k}^{(l)} = (-1)^{\nu+1},$$

$$\gamma_{m+1}^{(l)} \text{ and } \theta_{m+1}^{(l)} = \beta_k - \alpha_k.$$

After each iteration not one row, as in § 5, but two rows, $(\alpha, \beta)^{(l)}$ and $\Delta^{(l)}$, are added to the auxiliary tableau. In the upper part of the auxiliary tableau it is advisable to introduce the row α_j/β_j which gives the lower and the upper boundary values of the corresponding variables. When the l-th iteration is completed, the principal part of the l-th principal tableau and the l-th pair of rows in the lower part of the auxiliary tableau are filled.

The vector A_k to be introduced into the basis is determined as in the first algorithm. The coefficients in the expansion of the vector A_k in terms of the basis vectors and the vector A_{s_r} to be eliminated from the basis are determined in the same way as in the second algorithm for problems in canonical form. However, unlike the case of canonical form, the possibility $r=m+1$ may arise.

The coefficients $e_{ij}^{(l)}$ of the unit vectors when expanded in terms of the basis vectors $(i=1, 2, \ldots, m;\ j=1, 2, \ldots, m)$ are transformed according to recurrence formulas (1.21) if $\theta_0^{(l)}$ is obtained for $r \leqslant m$, and remain unmodified if $r=m+1$.

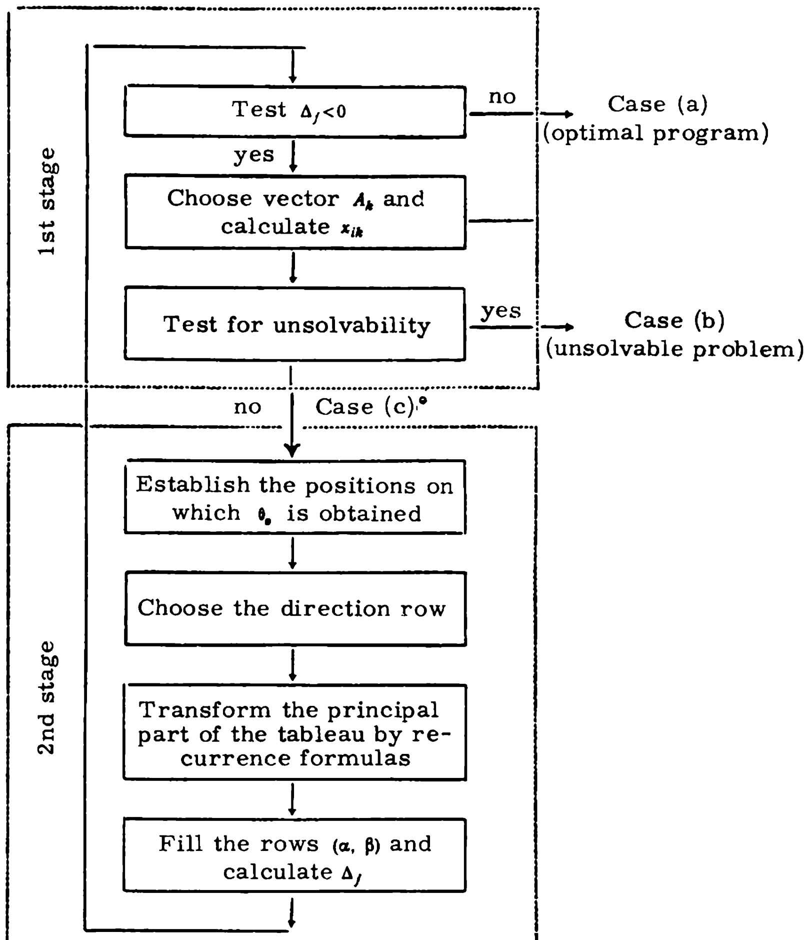


FIGURE 5.4

The basis components of the program, i.e., the entries of the column $e_0 = B - \sum_{j \in I_X} x_j A_j$ are transformed in the same way as the entries of the column

A_0 in the first algorithm:

$$e_{i0}^{(l+1)}=\begin{cases} e_{i0}^{(l)}-\theta_0^{(l)}\tilde{x}_{ik} & \text{for } i\neq r, \\ e_{m+1,0}^{(l)}-\theta_0^{(l)}(-1)^{v+1} & \text{for } i=r, \end{cases} \tag{7.7}$$
$$i=1,\ 2,\ \ldots,\ m.$$

Formulas (7.7) apply regardless of whether the basis is modified upon transition to the new program or not. The same recurrence formulas are used to transform the value $e_{m+1,0}^{(l)}$ of the linear form. To this end, in the $(m+2)$-th position of the column $\tilde{A}_k$ we write the evaluations

$$\tilde{x}_{m+2,k}^{(l)}=\Delta_k^{(l)}$$

of the vector introduced into the basis.

The rows $(\alpha,\ \beta)$ of the auxiliary tableau are transformed following the same rules as in the first algorithm. The parameters $\lambda_j^{(l)}$ are computed from recurrence formulas (1.21) with $i=m+1,\ j=1,\ 2,\ \ldots,\ n.$

The parameters $\Delta_j^{(l)}$ are computed from the formulas

$$\Delta_j^{(l)}=\begin{cases} \sum_{i=1}^{m} a_{ij}\lambda_i^{(l)}-c_j & \text{for } x_j=\alpha_j, \\ c_j-\sum_{i=1}^{m} a_{ij}\lambda_i^{(l)} & \text{for } x_j=\beta_j. \end{cases} \tag{7.8}$$

To unify the procedure we suggest that the difference $\sum_{i=1}^{m} a_{ij}\lambda_i^{(l)}-c_j$ be computed without taking into account the actual values of the extrabasis variables, and then, when examining the row $(\alpha,\ \beta)$, the signs of the entries for which $x_j=\beta_j$ should be changed.

Figure 5.4 shows a block diagram of the second algorithm for a problem with bilateral restraints.

7-4. We now deal with the specific features of the procedure for chosing the vector to be eliminated from the basis in the case of degenerate problems with bilateral restraints.

When solving the problem according to the first algorithm, it is advisable to define R_i thus:

$$R_i=\begin{cases} A_{s_i} & \text{for } x_{i0}<\beta_{s_i}, \\ -A_{s_i} & \text{for } x_{i0}=\beta_{s_i}. \end{cases} \tag{7.9}$$

When the second algorithm is applied, the following system of vectors R_i should be used:

$$\begin{gathered} R_1=A_XW;\quad R_2=e_1,\ \ldots,\ R_t=e_{t-1}, \\ R_{t+1}=e_t,\ \ldots,\ R_m=e_m. \end{gathered} \tag{7.10}$$

Here e_i are the m-dimensional unit vectors; the components of the vector W are determined by the formula

$$w_i=\begin{cases} 1 & \text{for } x_{i0}<\beta_{s_i}, \\ -1 & \text{for } x_{i0}=\beta_{s_i}. \end{cases}$$

A_X is the basis matrix, and the index t is chosen so that the t-th coefficient of the vector A_XW is nonzero.

It can be easily verified that the system (7.9), and likewise (7.10), satisfy the two conditions imposed on R_i in § 6 of the preceding chapter.

7-5. We shall now illustrate the sequence of computations leading to an optimal program of a linear-programming problem with bilateral restraints.

Maximize the linear form

$$L(X)=x_1+2x_2+3x_3+x_4+2x_5+3x_6+x_7-x_8-x_9-x_{10}-x_{11}$$

subject to the conditions

$$\begin{aligned}
x_1+\ x_2+2x_3+3x_4+\ \ 2x_5+3x_6+\ x_7+x_8 \qquad\qquad &=7,\\
2x_1+3x_2+\ x_3+\ x_4+\ \ 3x_5+2x_6+2x_7 \quad +x_9 \qquad &=8,\\
x_1+2x_2+3x_3+2x_4+0.5x_5+\ x_6+\ x_7 \qquad +x_{10} \quad &=6,\\
2x_1+\ x_2+3x_3+\ x_4+\ \ 2x_5+3x_6+\ x_7 \qquad\qquad +x_{11} &=7,\\
0\leqslant x_j\leqslant 1,\quad j=1,\ 2,\ \ldots,\ 7,&\\
x_j\geqslant 0,\quad j=8,\ \ldots,\ 11.&
\end{aligned}$$

The entire solution process of the problem by the second simplex algorithm is summarized in Tables 5.14 (the principal tableaus) and in Table 5.15 (the auxiliary tableau).

The unit basis $(A_8,\ A_9,\ A_{10},\ A_{11})$ is, obviously, taken as the initial basis. The corresponding support program has basis components equal to the components of the restraint vector, and zero extrabasis components. The column e_0 of the principal zeroeth tableau, in which the components of the vector

$$A_0=B-\sum_{j\in I_X} A_jx_j,$$

are written will, therefore, only contain components of the vector B.

The first $m=4$ entries of the columns $e_1,\ \ldots,\ e_4$ of the zeroeth tableau constitute a unit matrix. The $(m+1)$-th, 5th, row is filled after the vector to be introduced into the basis has been determined. The $(m+2)$-th, 6th, row contains the evaluations λ_j of the problem restraints and the value of the linear form. Thus,

$$L(X)=\lambda_0=-1\cdot 7-1\cdot 8-1\cdot 6-1\cdot 7=-28.$$

The other $\lambda_j\,(j=1,\ \ldots,\ 4)$ are equal to the linear-form coefficients associated with the corresponding basis variables.

In the row $(\alpha,\ \beta)$ of the lower part of the auxiliary tableau corresponding to the principal zeroeth tableau we indicate that in the initial support program all the extrabasis variables assume their lowest boundary value (α_j) (when filling the auxiliary tableau, we need indicate only which of the boundary values the extrabasis variables assume; the boundary value is not used in determining the sign of Δ_j). In our case, all the extrabasis variables vanish (being equal to the lowest boundary value), and the evaluations Δ_j are computed from the first of equations (7.8). The least $\Delta_j\,(-12)$ is obtained on the vectors A_3 and A_6. Either of these vectors can be introduced into the basis.

Let us choose the vector A_3. In column A_k of the principal zeroeth tableau we write the coefficients in the expansion of A_3 in terms of the basis vectors:

$$x_{m+1,\,3}=x_{5,\,3}=-1,\quad x_{m+2,\,3}=\Delta_3=-12.$$

The column $\tilde{A}_k$ coincides with the column A_k, since all the extrabasis variables assume their lowest boundary values. The first $m=4$ positions of the column γ are filled with zeros. The parameter

$$\gamma_{m+1}=\gamma_5=\beta_3=1,$$

since $x_3=\alpha_3$. The θ-values are computed from (7.2). For example,

$$\theta_2=\frac{x_{20}-\gamma_{20}}{\tilde{x}_{23}}=\frac{8-0}{1}=8,$$

$$\theta_{m+1}=\theta_5=\frac{x_{50}-\gamma_{50}}{\tilde{x}_{25}}=\frac{0-1}{-1}=1.$$

In our case $\theta_{m+1}=\theta_5$ is the least number in the column θ. Therefore, when passing to the next 1-th tableau, the basis remains unchanged.

The variable x_3 corresponding to the vector A_3 assumes, on the new program, its highest boundary value (and not the lowest as assumed in the previous program). The sign of Δ_3 will change accordingly. All the other extrabasis variables retain their original values, and the corresponding vectors retain their respective evaluations. We see from the auxiliary tableau that now the least evaluation corresponds to the vector $A_6\,(\Delta_6'=-12)$. The vector A_6 is introduced into the basis.

The column e_0 of the principal 1-th tableau contains the components of the vector

$$A_0=B-\sum_{j\in I_X} x_j'A_j.$$

In our case

$$e_0=B-\beta_3A_3=B-A_3.$$

TABLE 5.14 (0–7)

	No.	c_X	B_X	e_0	e_1	e_2	e_3	e_4	A_k	$\tilde{A}_k$	γ	θ	No. of tableau
	1	−1	A_8	7	1				2	2	0	3.5	0
	2	−1	A_9	8		1			1	1	0	8	
	3	−1	A_{10}	6			1		3	3	0	2	
	4	−1	A_{11}	7				1	3	3	0	2.333	
←	5	3	A_3	0	—	—	—	—	−1	−1	1	**1**	
	6	—	—	−28	−1	−1	−1	−1	−12	—	—	—	
	1	−1	A_8	5	1				3	3	0	1.667	1
	2	−1	A_9	7		1			2	2	0	3.5	
	3	−1	A_{10}	3			1		1	1	0	3	
	4	−1	A_{11}	4				1	3	3	0	1.333	
←	5	3	A_6	0	—	—	—	—	−1	−1	1	**1**	
	6	—	—	−16	−1	−1	−1	−1	−12	—	—	—	
	1	−1	A_8	2	1				2	2	0	1	2
	2	−1	A_9	5		1			3	3	0	1.667	
	3	−1	A_{10}	2			1		0.5	0.5	0	4	
←	4	−1	A_{11}	1				1	2	2	0	**0.5**	
	5	2	A_5	0	—	—	—	—	−1	−1	1	1	
	6	—	—	−4	−1	−1	−1	−1	−9.5	—	—	—	

TABLE 5.14 (continued)

	No.	C_X	B_X	e_0	c_1	e_2	e_3	e_4	A_k	$\tilde{A}_k$	γ	θ	No. of tableau
	1	−1	A_8	1	1			−1			—	—	3
	2	−1	A_9	3.5		1		−1.5	1.5	1.5	0	2.333	
	3	−1	A_{10}	1.75			1	−0.25	1.75	1.75	0	1	
→	4	2	A_5	0.5				0.5	0.5	0.5	0	2	
←	5	2	A_2	0	—	—	—	—	−1	−1	1	1	
	6	—	—	0.75	−1	−1	−1	3.75	−4.25	—	—	—	
	1	−1	A_8	1	1			−1	2	2	0	0.5	4
	2	−1	A_9	2		1		−1.5	−0.5	−0.5	—	—	
←	3	−1	A_{10}				1	−0.25	1.75	1.75	0	0	
	4	2	A_5					0.5	0.5	0.5	0	0	
	5	1	A_4	0	—	—	—	—	−1	−1	1	1	
	6	—	—	5	−1	−1	−1	3.75	−3.25	—	—	—	
←	1	−1	A_8	1	1		−1.143	−0.714	−3.571	3.571	0	0.28	5
	2	−1	A_9	2		1	0.286	−1.571	−2.857	2.857	0	0.700	
→	3	1	A_4				0.571	−0.143	1.286	−1.286	1	0.778	
	4	2	A_5				−0.286	0.571	0.857	−0.857	1	1.167	
	5	3	A_3	1	—	—	—	—	−1	1	0	1	
	6	—	—	5	—	—	0.857	3.286	−6.429	—	—	—	

TABLE 5.14 (continued)

	No.	C_X	B_X	e_0	e_1	e_2	e_3	e_4	A_k	$\tilde{A}_k$	γ	θ	No. of tableau
→	1	3	A_3	0.72	−0.28		0.32	0.2	0.08	−0.08	1	3.5	6
←	2	−1	A_0	1.20	−0.80	1	1.20	−1	−2.20	2.20	0	**0.545**	
	3	1	A_4	0.36	0.36		0.16	−0.4	0.04	−0.04	1	16	
	4	2	A_5	0.24	0.24		−0.56	0.4	1.36	−1.36	1	0.559	
	5	3	A_6	1	—	—	—	—	−1	1	0	1	
	6	—	—	6.8	0.80	−1	−1.20	2	−2.2	—	—	—	
	1	3	A_3	0.764	−0.309	0.036	0.364	0.164					7
→	2	3	A_6	0.455	0.364	−0.455	−0.545	0.455					
	3	1	A_4	0.382	0.345	0.018	0.182	−0.418					
	4	2	A_5	0.982	−0.255	0.618	0.182	−0.218					
	5				—	—	—	—					
	6	—	—	8				1		—	—	—	

TABLE 5.15

No.	B	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}
1	7	1	1	2	3	2	3	1	1			
2	8	2	3	1	1	3	2	2		1		
3	6	1	2	3	2	0.5	1	1			1	
4	7	2	1	3	1	2	3	1				1
5	C	1	2	3	1	2	3	1	−1	−1	−1	−1
0	(α, β)	α	α	α	α	α	α	α	—	—	—	—
	Δ	−7	−9	**−12**	−8	−9.5	−12	−6	—	—	—	—
1	$(\alpha, \beta)'$	α	α	β	α	α	α	α	—	—	—	—
	Δ'	−7	−9	12	−8	−9.5	**−12**	−6	—	—	—	—

TABLE 5.15 (continued)

No.	B	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}
2	$(\alpha, \beta)''$	α	α	β	α	α	β	α	—	—	—	—
	Δ''	−7	−9	12	−8	**−9.5**	12	−6	—	—	—	—
3	$(\alpha, \beta)'''$	α	α	β	α	—	β	α	—	—	—	α
	Δ'''	2.5	−4.25	−2.25	−3.25	—	−2.25	−1.25	—	—	—	4.75
4	$(\alpha, \beta)^{IV}$	α	β	β	α	—	β	α	—	—	—	α
	Δ^{IV}	2.5	4.25	−2.25	**−3.25**	—	−2.25	−1.25	—	—	—	4.75
5	$(\alpha, \beta)^{V}$	α	β	β	—	—	β	α	—	—	α	α
	Δ^{V}	3.429	1	**6.429**	—	—	−2.714	0.143	—	—	1.857	4.286
6	$(\alpha, \beta)^{VI}$	α	β	—	—	—	β	α	α	—	α	α
	Δ^{VI}	0.6	4.6	—	—	—	−2.2	−1.4	1.8	—	−0.2	3
7	$(\alpha, \beta)^{VII}$	α	β	—	—	—	—	α	α	α	α	α
	Δ^{VII}	1	1	—	—	—	—	0	1	1	1	2

Thus, when passing from the zeroeth tableau to the 1st tableau the entire principal part of the tableau, except for the column A_0 (entries $1, 2, \ldots, m, m+2$), remains uncharged. The columns A_k, $\bar{A}_k$, γ, and θ and the $(m+1)$-th row are filled as in the preceding tableau. The minimum is again obtained in the $(m+1)$-th entry. Hence, in this iteration, too, transformation to a new support program will not affect the basis.

In the next iteration (2nd tableau) the vector A_6 is introduced into the basis in place of the unit vector e_{lj}. In this case the principal part of the tableau is transformed according to the recurrence formulas (the basis components of the support program and the value of the linear form are computed from (7.7), and the elements A_{il} of the inverse matrix and the relative evaluations λ_j of the problem restraints from (1.21)).

After six iterations, all the evaluations Δ_j of the restraint vectors become nonnegative. The optimal basis comprises the vectors A_3, A_4, A_5, A_6. The basis components of the solution are

$$x_3 = 0.764, \quad x_4 = 0.382, \quad x_5 = 0.982, \quad x_6 = 0.455.$$

Of all the extrabasis variables only $x_2 = 1$ is nonzero.

§ 8. Computational procedure for determining the support program

8-1. In discussing the simplex algorithms, we invariably assumed the initial support program to be known.

In § 7 of the preceding chapter we gave some theoretical considerations on the possibility of constructing an initial support program and described procedures for combining two stages, namely computation of the initial program and solution of the problem. In the present chapter we will deal with the realization of the corresponding computational procedures.

First, consider a method of computing the initial support program which amounts to solving the so-called auxiliary problem (see Chapter 4, § 7). To solve the auxiliary problem, we may, generally speaking, apply either the first or the second algorithm. In some cases, however, application of the first algorithm entails less computations. The reason for this is that the optimal basis of the auxiliary problem often contains, besides restraint vectors of the principal problem, also the unit vectors e_i (artificial vectors). To replace the unit vectors by restraint vectors, we must know the coefficients in the expansion of all the vectors A_j in terms of respective basis vectors. The coefficients x_{ij} for all j are computed according to the first algorithm only.

When solving the auxiliary problem, we need not compute the coefficients in the expansion of the unit vectors in terms of the basis vectors. The unit vectors constitute the initial basis of the auxiliary problem. However, when solving the problem, it is not advisable to include artificial vectors in the basis. Therefore, we may omit from the tableau of the algorithm all the columns corresponding to the unit restraint vectors of the auxiliary problem.

The process of solution of the auxiliary problem, whose initial program is obvious, terminates after a finite number of iterations in case (a) (case (b) is ruled out, since the linear form of the auxiliary problem is zero-bounded above). Two possibilities may arise. If the maximum value of the linear form is zero, the basis variables of the auxiliary problem specify the initial support program of the principal problem. If not all the artificial variables vanish in the solution of the auxiliary problem, the principal problem is unsolvable.

Let the optimal value of the linear form of the auxiliary problem be zero.

As we have already indicated, the optimal basis may contain additional unit vectors. The corresponding basis variables all, obviously, vanish. We shall show, without modifying the program, how to replace the unit basis vectors by restraint vectors, thus complementing the basis to the maximum system of linearly independent restraint vectors of the principal problem. If the maximum system contains m vectors, the rank of the restraint system of the problem is also m. If the maximum system contains $r<m$ vectors, then, by suitable transformation of the basis, we shall isolate r linearly independent equalities – problem restraints.

Consider the tableau corresponding to the last iteration of the solution process of the auxiliary problem. Two cases are possible:

(a) all the elements x_{ij} of the principal part of the tableau in all the rows corresponding to the additional unit vectors are zero;

(b) the optimal basis of the auxiliary problem contains an additional vector e_i for which at least one of the coefficients x_{ij} is not zero.

In (a) the restraint vectors of the principal problem entering the optimal basis of the auxiliary problem constitute the maximum system of linearly independent vectors of the matrix $A=\|a_{ij}\|_{mxn}$. Indeed, these vectors belong to the basis and are, therefore, linearly independent. Moreover, any restraint vector can be expanded in terms of the basis vectors so that the coefficients in the expansion corresponding to the additional unit vectors are zero. Hence, all the vectors A_j can be expanded in terms of the restraint vectors of the principal problem entering the optimal basis of the auxiliary problem.

We now consider (b). Let the i-th position of the basis of the auxiliary problem contain an additional unit vector e_i and let $x_{ip}\neq 0$. We substitute restraint vector A_p for the basis vector e_i. The resulting system of vectors remains linearly independent and, as we see from recurrence formulas (1.12), the elements of column A_0 retain their previous values. The number of additional unit vectors in the basis of the optimal program of the auxiliary problem is thus reduced by one, whereas the program is not modified in the procedure. Similarly, the restraint vectors of the principal problem can be successively substituted for all the additional unit vectors to each of which correspond at least one nonzero coefficient x_{ij}. The maximum system of restraint vectors will contain m vectors, if all the additional unit vectors are removed from the basis of the optimal program of the auxiliary problem. The maximum system contains $r<m$ vectors, if after substitution we obtain a tableau in which all the elements of the principal part corresponding to the $m-r$ unit basis vectors are zero. In such a case, the tableau rows corresponding to the unit artificial vectors should be omitted. The construction of an optimal program of the principal problem is analyzed with the aid of an $r\times n$ restraint matrix. Thus, in the process of solution of the auxiliary problem, an initial support program is determined and the independent restraints of the principal linear-programming problem are isolated.

8-2. We shall illustrate the above by a simple example.

Determine the initial support program for the problem of maximization of some linear form $L(X)$ subject to the conditions

$$
\begin{gathered}
2x_1+3x_2-x_3+4x_4+x_5=8,\\
3x_1+2x_2+2x_3-2x_4-2x_5=7,\\
5x_1+5x_2+x_3+x_4-x_5=15,\\
x_1+4x_2-4x_3+11x_4+4x_5=9,\\
x_j\geqslant 0,\quad j=1, 2, \dots, 5.
\end{gathered}
$$

TABLE 5.16 (0–2)

c_X	B	A_0	A_1	A_2	A_3	A_4	A_5	θ	No. of tableau
−1	A_6	8	2	3	−1	4	−1	$^8/_3$	
−1	A_7	7	3	2	2	−2	−2	$^7/_2$	
−1	A_8	15	5	5	1	1	−1	3	0
−1	A_9	9	1	4	−4	11	4	$^9/_4$	
—	—	−39	−11	−14	2	−13	−2	—	
−1	A_6	$^5/_4$	$^5/_4$		2	$-^{17}/_4$	−2	$^5/_8$	
−1	A_7	$^5/_2$	$^5/_2$		4	$-^{15}/_2$	−4	$^5/_8$	
−1	A_8	$^{15}/_4$	$^{15}/_4$		6	$-^{51}/_4$	−6	$^5/_8$	1
	A_2	$^9/_4$	$^1/_4$	1	−1	$^{11}/_4$	1	—	
—	—	$-^{15}/_2$	$-^{15}/_2$		−12	$^{51}/_2$	12	—	
	A_3	$^5/_8$	$^5/_8$		1	$-^{17}/_8$	−1		
−1	A_7					1			
−1	A_8								2
	A_2	$^{23}/_8$	$^7/_8$	1		$^5/_8$			
—	—					−1		—	

TABLE 5.17

B	A_0	A_1	A_2	A_3	A_4	A_5
A_3	$^5/_8$	$^5/_8$		1		−1
A_4					1	
A_8						
A_2	$^{23}/_8$	$^7/_8$	1			

The auxiliary problem involves maximization of the linear form

$$\tilde{L}(X) = -(x_6 + x_7 + x_8 + x_9)$$

subject to the conditions

$$\begin{aligned} &L_1(X) + x_6 &&= 8, \\ &L_2(X) \quad + x_7 &&= 7, \\ &L_3(X) \quad\quad + x_8 &&= 15, \\ &L_4(X) \quad\quad\quad + x_9 &&= 9, \\ &x_j \geqslant 0, \quad j = 1, 2, \ldots, 9. \end{aligned}$$

Here $L_i(X)$ are the left-hand sides of the restraints of the initial problem. We shall solve the auxiliary problem applying the first simplex algorithm. The solution process is obvious from Table 5.16. As pointed out it is unnecessary to include columns in the tableaus for the artificial vectors.

The optimal basis of the auxiliary problem contains two artificial vectors A_7 and A_8. We see from the 2nd tableau that the coefficient $x_{24}^{(2)}$ in the expansion of the vector A_4 in terms of the vectors of the last basis is nonzero. The artificial basis vector A_7 can, therefore, be replaced by the restraint vector A_4. Having transformed the last tableau, we obtain Table 5.17. The elements of the row A_8 are all zeros. Hence, the vectors A_3, A_4, and A_2 constitute the maximum system of linearly independent vectors of the restraint matrix.

The optimal value of the linear form of the auxiliary problem is zero. Therefore, the solution of the auxiliary problem specifies the initial support program of the principal problem. Moreover, we see from the last tableau that the linearly independent restraints of the principal problem can be rewritten in the form

$$\begin{aligned} &\frac{5}{8}x_1 \quad + x_3 \quad - x_5 = \frac{5}{8}, \\ &\qquad\qquad x_4 \qquad = 0, \\ &\frac{7}{8}x_1 + x_2 \qquad\quad = \frac{23}{8}. \end{aligned}$$

Thus, solving the auxiliary problem we obtain not only an initial program of the problem, $X_0 = \left(0, \frac{23}{8}, \frac{5}{8}, 0, 0\right)$, but also a simplified form of the problem restraints.

8-3. We will now see how to use the M-method which makes it possible to combine the determination of the initial support program with the solution of the linear-programming problem. The optimal program of the M-problem, whose initial program is obvious, can be computed either according to the first or to the second simplex algorithm.

The linear-form coefficients c_j of the M-problem are linear functions of the parameter M:

$$c_j = \bar{c}_j M + \bar{\bar{c}}_j.$$

The parameters Δ_j and λ_i are, therefore, also linear functions of M:

$$\Delta_j = \bar{\Delta}_j M + \bar{\bar{\Delta}}_j, \ \bar{\lambda}_i = \bar{\lambda}_i M + \bar{\bar{\lambda}}_i.$$

Hence the specific features of the tableaus in the computational procedure of the M-method. Solving the M-problem, according to the first or the second algorithm, not only one row, Δ, in the tableaus is filled but two rows, the $\bar{\Delta}$ row of the coefficients of M and the $\bar{\bar{\Delta}}$ of the free terms. The evaluations Δ_j of the vectors A_j are compared as follows:

$$\Delta_{j_1} > \Delta_{j_2}, \quad \text{if} \quad \begin{cases} \bar{\Delta}_{j_1} > \bar{\Delta}_{j_2}, \\ \bar{\Delta}_{j_1} = \bar{\Delta}_{j_2}, \text{ but } \bar{\bar{\Delta}}_{j_1} > \bar{\bar{\Delta}}_{j_2}. \end{cases}$$

Correspondingly, the evaluation Δ_j is considered positive if either $\bar{\Delta}_j > 0$ or $\bar{\Delta}_j = 0$, but $\bar{\bar{\Delta}}_j > 0$.

Computing according to the second algorithm, the last row of the principal tableaus (the row of the λ-evaluations of the problem restraints with respect to the current basis) is also replaced by two rows, $\bar{\lambda}$ and $\bar{\bar{\lambda}}$.

The artificial vectors eliminated from the basis are not considered in the solution of the M-problem. The additional unit vectors are, therefore,

not included in the columns of the restraint vector in the tableaus in which the solution of the M-problem is written. It can easily be seen that this rule reduces the volume of computations, without essentially affecting the M-method.

The solution process of the M-problem terminates, after a finite number of iterations, in case (a) or case (b). In case (a) the solution of the M-problem specifies an optimal program of the principal problem, if the additional variables do not enter the solution. All other possibilities mean unsolvability of the principal problem (the linear form is unbounded in the set of feasible programs of the problem or the problem restraints are inconsistent).

Apart from the above special features, the process of solution of the M-problem according to the first or second simplex algorithms is the same as that discussed in § 2 and § 5.

8-4. We now illustrate the application of the M-method to the solution of a linear-programming problem whose initial program is not obvious. The computations will be carried out according to the second simplex algorithm.

Maximize the linear form

$$L(X)=x_1+2x_2+3x_3+x_4+2x_5+3x_6+x_7 \tag{8.1}$$

subject to the conditions

$$\left.\begin{aligned} x_1+x_2+2x_3+3x_4+2x_5+3x_6+x_7&=7,\\ 2x_1+3x_2+x_3+x_4+3x_5+2x_6+2x_7&=8,\\ x_1+2x_2+3x_3+2x_4+0,5x_5+x_6+2x_7&=6,\\ 2x_1+x_2+3x_3+x_4+2x_5+3x_6+2x_7&=7, \end{aligned}\right\} \tag{8.2}$$

$$x_j\geqslant 0,\quad j=1, 2, \ldots, 7.$$

No initial support program is given. We introduce artificial nonnegative variables x_8, x_9, x_{10}, and x_{11} and consider the corresponding M-problem.

Maximize the linear form

$$\bar{L}=L(X)-M(x_8+x_9+x_{10}+x_{11})$$

subject to the conditions

$$\begin{aligned} &L_1(X)+x_8 &&=7,\\ &L_2(X)\quad +x_9 &&=8,\\ &L_3(X)\qquad +x_{10} &&=6,\\ &L_4(X)\qquad\quad +x_{11}&&=7,\\ &x_j\geqslant 0,\quad j=1, 2, \ldots, 11. \end{aligned}$$

Here $L(X)$ is the linear form (8.1) and $L_i(X)$ represents the left-hand sides of restraints (8.2) of the principal problem.

We give here the principal and the auxiliary tableaus (Tables 5.18 and 5.19) which illustrate the successive program improvement according to the second algorithm.

The tableaus do not need special explanations. The solution is obtained in four iterations. The optimal program is

$$X=\left\{0, \frac{25}{16}, \frac{3}{8}, \frac{3}{16}, 0, \frac{11}{8}, 0\right\}.$$

The maximum value of the linear form is 137/16.

In this example the optimal program is obtained directly when all the artificial vectors have been eliminated from the basis. In the general case, however, this need not be so. After the artificial vectors are eliminated from the basis, we usually obtain a support program which is then transformed by the simplex procedure into an optimal program.

8-5. We shall now show how computation of the initial support program can be simplified in certain cases.

Until now we associated the auxiliary problem and the M-problem with a complete artificial basis, namely with m additional unit vectors. The

TABLE 5.18 (0–4)

	No.	C_X	B_X	e_0	e_1	e_2	e_3	e_4	A_k	θ	No. of tableaus
	1	$-M$	A_8	7	1				2	$7/2$	0
	2	$-M$	A_9	8		1			1	8	
←	3	$-M$	A_{10}	6			1		3	2	
	4	$-M$	A_{11}	7				1	3	$7/3$	
	5	—	—	-28	-1	-1	-1	-1	-9	—	
									-3		
	1	$-M$	A_8	3	1		$-2/3$		$7/3$	$9/7$	1
	2	$-M$	A_9	6		1	$-1/3$		$5/3$	$18/5$	
→	3	3	A_3	2			$1/3$		$1/3$	6	
←	4	$-M$	A_{11}	1			-1	1	2	$1/2$	
	5	—	—	-10	-1	-1	2	-1	-6	—	
				6			1		-2		
	1	$-M$	A_8	$11/6$	1		$1/2$	$-7/6$	$5/6$	$11/5$	2
←	2	$-M$	A_9	$31/6$		1	$1/2$	$-5/6$	$19/6$	$31/19$	
	3	3	A_3	$11/6$			$1/2$	$-1/6$	$5/6$	$11/5$	
→	4	3	A_6	$1/2$			$-1/2$	$1/2$	$-1/2$	—	
	5	—	—	-7	-1	-1	-1	2	-4	—	
				7				1	-1		
←	1	$-M$	A_8	$9/19$	1	$-5/19$	$7/19$	$-18/19$	$48/19$	$9/48$	3
→	2	2	A_2	$31/19$		$6/19$	$3/19$	$-5/19$	$7/19$	$31/7$	
	3	3	A_3	$9/19$		$-5/19$	$7/19$	$1/19$	$10/19$	$9/10$	
	4	3	A_6	$25/19$		$3/19$	$-8/19$	$7/19$	$-6/19$	—	
	5	—	—	$-9/19$	-1	$5/19$	$-7/19$	$18/19$	$-48/19$	—	
				$164/19$		$6/19$	$8/19$	$14/19$	$7/19$		
→	1	1	A_4	$3/16$	$19/48$	$-5/48$	$7/48$	$-3/8$			4
	2	2	A_2	$25/16$	$-7/48$	$17/48$	$5/48$	$-1/8$			
	3	3	A_3	$3/8$	$-5/24$	$-5/24$	$7/24$	$1/4$			
	4	3	A_6	$11/8$	$1/8$	$1/8$	$-3/8$	$1/4$			
	5	—	—							—	
				$137/16$	$-7/48$	$17/48$	$5/48$	$7/8$			

TABLE 5.19

No.	B	A_1	A_2	A_3	A_4	A_5	A_6	A_7
1	7	1	1	2	3	2	3	1
2	8	2	3	1	1	3	2	2
3	6	1	2	3	2	0,5	1	2
4	7	2	1	3	1	2	3	2
5	C	1	2	3	1	2	3	1
0	Δ	−6	−7	$\boxed{-9}$	−7	−7,5	−9	−7
		−1	−2	$\boxed{-3}$	−1	−2	−3	−1
1	Δ'	−3	−1		−1	−6	$\boxed{-6}$	−1
					1	−1,5	$\boxed{-2}$	1
2	Δ''		$\boxed{-4}$		−4	−1,5		−1
		1	$\boxed{-1}$					1
3	Δ'''	$^{20}/_{19}$			$\boxed{-^{48}/_{19}}$	$^{1}/_{2}$		$^{13}/_{19}$
		$^{24}/_{19}$			$\boxed{^{7}/_{19}}$	$^{1}/_{2}$		$^{27}/_{19}$
4	Δ^{IV}							
		$^{17}/_{12}$				$^{55}/_{96}$		$^{73}/_{48}$

bulkiness of computations is considerably reduced if the principal problem has some unit restraint vectors. In several cases simple manipulations enable us to reduce the number of artificial variables of the auxiliary problem or the M-problem.

Let the problem restraints be given in the form

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i,$$
$$i=1, 2, \ldots, m; \quad x_j \geqslant 0, \quad j=1, 2, \ldots, n; \quad b_i \geqslant 0.$$

As we have remarked (Chapter 5, § 7), the additional nonnegative variables x_{n+i} $(i=1, 2, \ldots, m)$ reduce the problem to canonical form. The values of the additional variables $x_{n+i}=b_i$ specify the initial support program.

Now let the problem restraints have the form

$$\sum_{j=1}^{n} a_{ij}x_j \geqslant b_i,$$
$$i=1, 2, \ldots, m; \quad x_j \geqslant 0, \quad j=1, 2, \ldots, n; \quad b_i \geqslant 0.$$

The additional variables x_{n+i} $(i=1, 2, \ldots, m)$ will again enable us to reduce the problem to canonical form. However, to obtain the initial support program we must, in this case, solve an auxiliary problem (or the M-problem) with one artificial vector. We have

$$\sum_{j=1}^{n} a_{ij}x_j - x_{n+i} = b_i,$$
$$i=1, 2, \ldots, m, \quad x_j \geqslant 0, \quad j=1, 2, \ldots, n+m.$$

Let $b_s = \max_i b_i$. We transform the system of restraints to the form

$$\sum_{j=1}^{n} (a_{sj}-a_{ij})\, x_j - x_{n+s} + x_{n+i} = b_s - b_i, \qquad i \neq s,$$

$$\sum_{j=1}^{n} a_{sj}x_j - x_{n+s} = b_s, \quad x_j \geqslant 0, \quad j=1, 2, \ldots, n+m.$$

Thus, a problem where $m-1$ positive unit restraint vectors are associated with the nonnegative components of the constraint vector is obtained. Determination of the initial support program thus involves one artificial vector.

Finally, let the problem restraints be given in canonical form:

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \qquad i=1, 2, \ldots, m, \quad x_j \geqslant 0, \quad j=1, 2, \ldots, n.$$

Let all $b_i \geqslant 0$ but at least one of the components of the constraint vector be positive.

There are three possibilities:

(a) there are no unit vectors among the restraint vectors; in this case the M-problem (or the auxiliary problem) contains m artificial variables;

(b) the matrix $\|a_{ij}\|$ contains r different unit restraint vectors; in this case the M-problem contains $m-r$ artificial vectors.

(c) the restraint matrix $\|a_{ij}\|$ contains a unit submatrix of order m; the initial program here is obvious and there is no need to resort to the M-method or to solve the auxiliary problem.

In some linear-programming problems a system of linearly independent vectors can be isolated from among the restraint vectors. In these cases determination of the initial support program is greatly simplified. Let the vectors $A_1, A_2, \ldots, A_m$ be linearly independent. We solve the system of linear equations

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \qquad i=1, 2, \ldots, m,$$

for the variables $x_1, \ldots, x_m$. We have

$$x_i = x_i^0 + \sum_{j=m+1}^{n} x_{ij}^0 x_j,$$

where x_i^0 and x_{ij}^0 ($i=1, \ldots, m$; $j=1, \ldots, n$) are constants. If all $x_i^0 \geqslant 0$, then, taking $x_j = 0$ for $j = m+1, \ldots, n$, we obtain a support program $X = (x_1^0, \ldots, x_m^0, 0, \ldots, 0)$.

Let now some of the x_i^0 be negative, x_s^0 being the least. We subtract, term-by-term, the s-th equation from all the equations of the system. Then

$$x_i = (x_i^0 - x_s^0) + x_s + \sum_{j=m+1}^{n} (x_{ij}^0 - x_{sj}^0)\, x_j, \quad x_s = x_s^0 + \sum_{j=m+1}^{n} x_{sj}^0 x_j.$$

This transformation yields a basis comprising $m-1$ positive unit restraint vectors of the problem and one artificial vector.

§ 9. Cycling in linear-programming problems

9-1. In Chapter 4, § 6 we gave rules for determining the vector to be eliminated from the basis which guarantee that cycling will not arise when solving linear-programming problems. As we have seen, in §§ 2, 5, these rules involve additional computational work.

It is, therefore, natural that we should consider the possibility of cycling arising when a linear-programming problem is solved by the simplex method and the probability of this phenomenon. The present section deals with the conditions under which cycling arises; this discussion will provide the answers to the above queries. All reasoning refers to the problem in canonical form.

For the following we require recurrence formulas relating the parameters of two successive simplex iterations. The suitable formulas were derived in § 1. We repeat them here assuming that in the successive iteration the vector A_k is introduced into the r-th position in the basis:

$$x'_{ij} = \begin{cases} x_{ij} - \dfrac{x_{ik}}{x_{rk}} x_{rj}, & \text{if} \quad i \neq r, \\ \dfrac{x_{rj}}{x_{rk}}, & \text{if} \quad i = r, \end{cases} \tag{9.1}$$

$$i = 1, 2, \ldots, m, \quad j = 0, 1, 2, \ldots, n,$$

$$\Delta'_j = \Delta_j - \frac{\Delta_k}{x_{rk}} x_{rj}, \qquad j = 1, 2, \ldots, n. \tag{9.2}$$

Here, as elsewhere, the primed parameters refer to the successive iteration.

Applying formulas (9.1) and (9.2) twice, we can establish a relationship between the parameters for any two iterations with one iteration between them. Assume that in the second iteration the vector A_k is introduced into the r-th position, and in the third iteration the vector A_{k_1} is introduced into the r_1-th position. Then

$$x''_{ij} = \frac{x_{r_1 k_1} x_{rj} - x_{rk_1} x_{r_1 j}}{x_{r_1 k_1} x_{rk} - x_{r_1 k} x_{rk_1}}, \qquad i = r \neq r_1, \quad i = r = 1, 2, \ldots, m;$$

$$j = 0, 1, 2, \ldots, n; \tag{9.3}$$

$$\Delta''_j = \Delta_j - \Delta_k \frac{x_{r_1 k_1} x_{rj} - x_{rk_1} x_{r_1 j}}{x_{r_1 k_1} x_{rk} - x_{r_1 k} x_{rk_1}} - \Delta_{k_1} \frac{x_{rk} x_{r_1 j} - x_{r_1 k} x_{rj}}{x_{r_1 k_1} x_{rk} - x_{r_1 k} x_{rk_1}}, \quad r \neq r_1. \tag{9.4}$$

The double-primed parameters refer to the last iteration (of the three iterations in question).

Formulas (9.3) and (9.4) apply in cases when the vectors A_k and A_{k_1} are introduced into different basis positions, and in (9.3) the i-th position is assumed to coincide with one of them. In this section we shall consider this case only. Derivation of formulas (9.3) and (9.4) are left to the reader (see Exercise 11). Formula (9.3) can, obviously, be adapted to the case $i=r_1\neq r$. It suffices to substitute k_1 and r_1 for k and r, respectively.

9-2. Assume that the rule given in Chapter 4, § 6 is not applied when determining which vector is to be eliminated from the basis. Then, the choice of the basis position in which a new vector is to be introduced is equally simple in the nondegenerate and the degenerate case. Indeed, we choose the position r such that

$$\frac{x_{r0}}{x_{rk}}=\theta_0=\min_{x_{ik}>0}\frac{x_{i0}}{x_{ik}}. \tag{9.5}$$

Here k is the index of the restraint vector introduced into the basis, A_k. We recall that the vector A_k introduced into the basis has a negative evaluation Δ_k with respect to the original basis. In the degenerate case there are several basis positions satisfying condition (9.5); any of these can be chosen as r. We shall say that a vector A_k is *suitable* for introduction into the r-th position in the basis if $\Delta_k<0$ and condition (9.5) is satisfied.

We will study the possibility of *cycling*, i.e., the possibility of the following sequence of bases arising when solving a problem by the simplex procedure:

$$\mathrm{B}\to\mathrm{B}_1\to\mathrm{B}_2\to\ldots\to\mathrm{B}_{t-1}\to\mathrm{B} \tag{9.6}$$

We shall say that a *cycle* is of *length* d, if cycling comprises d iterations. The length of cycle (9.6) is thus t.

Obviously, transferring from basis to basis as in (9.6) should not increase the linear form of the problem, since otherwise the transition from B_{t-1} to B would be ruled out (the linear form cannot decrease!). Therefore, in all the iterations constituting cycle (9.6) all θ_0 are zero.

Hence, in particular, all the elements of (9.6) are bases of the same support program.

Let the basis B_{l+1} be obtained from B_l by introducing vector A_k into the r-th position. Then, in order to transfer from B_l to B_{l+1} it is required that the vector A_k be suitable for introduction into the r-th position. In this case $\theta_0=0$.

These conditions, therefore, can be written as

$$\Delta_k^{(l)}<0, \tag{9.7}$$

$$x_{r0}^{(l)}=0,\quad x_{rk}^{(l)}>0. \tag{9.8}$$

Inequality (9.7) shows that the evaluation of A_k with respect to B_l is negative; conditions (9.8) are equivalent to (9.5). The superscript (l) indicates that the parameters refer to the l-th basis B_l in chain (9.6) (the parameters associated with the basis B have no superscript).

In the following we shall also require an alternative form of conditions (9.7), (9.8) in which the parameters of the successive basis B_{l+1} appear.

For the vector A_k to be suitable for introduction into the r-th position of the basis upon transformation from B_l to B_{l+1}, it is necessary and

sufficient that

$$\Delta_{s_r}^{(l+1)} > 0, \tag{9.9}$$

$$x_{r0}^{(l+1)} = 0, \quad x_{rs_r}^{(l+1)} > 0. \tag{9.10}$$

Here s_r is the index of the vector A_{s_r} introduced into the r-th position of the basis B_l. To prove this proposition we apply the recurrence formulas (9.1) and (9.2).

Since $x_{rs_r}^{(l)} = 1$ and $\Delta_{s_r}^{(l)} = 0$, we obtain

$$x_{r0}^{(l+1)} = \frac{x_{r0}^{(l)}}{x_{rk}^{(l)}}, \tag{9.11}$$

$$x_{rs_r}^{(l+1)} = \frac{1}{x_{rk}^{(l)}}, \tag{9.12}$$

$$\Delta_{s_r}^{(l+1)} = -\frac{\Delta_k}{x_{rk}^{(l)}}. \tag{9.13}$$

The equivalence of conditions (9.8) and (9.10) follows from equalities (9.11) and (9.12). The equivalence of (9.7) and (9.9) follows from equality (9.13).

9-3. In dealing with the possibility of a chain (9.6) arising, we set out from the simplest case. We assume that in all the iterations constituting the chain new vectors are introduced in the same position in the basis. We shall first establish the following auxiliary proposition:

Lemma 9.1. Let B_{l+1} be obtained from B_l by introducing the restraint vector A_k into the r-th basis position. If the vector A_q is suitable to be introduced into the r-th position of the basis B_{l+1}, then it is also suitable to be introduced into the r-th position of the basis B_l.

Proof. By assumption,

$$\Delta_q^{(l+1)} < 0, \quad x_{r0}^{(l+1)} = 0, \quad x_{rq}^{(l+1)} > 0. \tag{9.14}$$

From recurrence formulas (9.1) and (9.2),

$$\Delta_q^{(l+1)} = \Delta_q^{(l)} - \frac{\Delta_k^{(l)}}{x_{rk}^{(l)}} x_{rq}^{(l)}, \tag{9.15}$$

$$x_{rj}^{(l+1)} = \frac{x_{rj}^{(l)}}{x_{rk}^{(l)}}. \tag{9.16}$$

From (9.16) for $j = 0, q$ and from the last two relationships in (9.14), we obtain

$$x_{r0}^{(l)} = x_{rk}^{(l)} x_{r0}^{(l+1)} = 0,$$
$$x_{rq}^{(l)} = x_{rk}^{(l)} x_{rq}^{(l+1)} > 0 \quad (x_{rk}^{(l)} > 0).$$

Further, since $\Delta_k^{(l)} < 0$, $x_{rk}^{(l)} > 0$, and $x_{rq}^{(l)} > 0$, we have

$$\Delta_q^{(l)} = \Delta_q^{(l+1)} + \frac{\Delta_k^{(l)}}{x_{rk}^{(l)}} x_{rq}^{(l)} < \Delta_q^{(l+1)} < 0.$$

The vector A_q thus satisfies conditions (9.7) and (9.8). This completes the proof.

Now, for any $l, 0 \leqslant l \leqslant t$, let

$$B_l = (A_{j_l}, A_2, \ldots, A_{m-1}, A_m), \tag{9.17}$$

where $A_{j_0} = A_{j_t} = A_1$, $B_0 = B$.

Theorem 9.1. A cycle (9.6) comprising bases of the form (9.17) cannot arise.

Proof of this theorem follows immediately from Lemma 9.1. Indeed, let us assume the contrary. Then the vector $A_1 = A_{l_t}$ is suitable to be introduced into the r-th basis position when transforming from B_{t-1} to $B_t = B$. Hence, from Lemma 9.1, the vector $A_1 = A_{l_t}$ is suitable to be introduced into the r-th basis position when transforming from B_{t-2} to B_{t-1}. Applying Lemma 9.1 in this way t times, we conclude that vector A_1 is suitable to be included into the r-th position of the basis when transforming from B to B_1. According to condition (9.7) this, in particular, indicates that the evaluation Δ_1 of the vector A_1 with respect to the basis $B = (A_1, A_2, \ldots, A_{m-1}, A_m)$ is negative. On the other hand $\Delta_1 = 0$, since the vector A_1 enters the basis B. This contradiction proves the theorem.

The number of zero components of the support program is called the **degree of degeneracy** (or simply **degeneracy**) of this program. It follows from Theorem 9.1 that cycle (9.6) may arise when the degree of degeneracy of the support program corresponding to the bases in chain (9.6) is greater than unity. The same conclusion can be drawn from the results of Chapter 4, § 6 (see Chapter 4, Exercise 8).

9-4. Theorem 9.1 shows that cycle (9.6) can arise only if new vectors are introduced into at least two basis positions. We will consider the simplest of all possible cases, i.e., we assume that new vectors in chain (9.6) are introduced into exactly two basis positions.

To be specific, let these positions be 1 and 2. In this case the basis B_t is completely defined by the two restraint vectors in the first two positions (the vectors occupying the other positions remain unchanged). Therefore, for any of the bases (9.6) we shall indicate only the vectors occupying the positions 1 and 2.

Consider the chain

$$B(A_1, A_2) \longrightarrow B_1(A_3, A_2) \longrightarrow B_2(A_3, A_4) \longrightarrow B_3(A_1, A_4) \longrightarrow B(A_1, A_2). \qquad (9.18)$$

Let us write the conditions under which this chain is possible. Since A_3 is introduced into the 1st position of the basis when transforming from B to B_1, we have from (9.7) and (9.8)

$$x_{13} > 0, \quad \Delta_3 < 0. \qquad (9.19)$$

In the next iteration, the vector A_4 is introduced into the 2nd position of the basis. Hence

$$x_{24}^{(1)} > 0, \quad \Delta_4^{(1)} < 0.$$

Expressing $x_{24}^{(1)}$ and $\Delta_4^{(1)}$ in terms of the parameters of B from recurrence formulas (9.1) and (9.2) ($k = 3$, $r = 1$), we have

$$\begin{aligned} x_{24}^{(1)} &= x_{24} - \frac{x_{23}}{x_{13}} x_{14} > 0, \\ \Delta_4^{(1)} &= \Delta_4 - \frac{\Delta_3}{x_{13}} x_{14} < 0. \end{aligned} \qquad (9.20)$$

Now take the fourth link in the chain. Here the vector A_2 is substituted for A_4 in the 2nd basis position, resulting in the initial basis B. We apply conditions (9.9) and (9.10). In this case $B_t = B_3$, $B_{t+1} = B$, $r = 2$, $s_r = 4$. Hence

$$x_{24} > 0, \quad \Delta_4 > 0. \qquad (9.21)$$

Finally, in the third link of chain (9.18) the vector A_3 occupying the 1st position is replaced by A_1. Again applying (9.9) and (9.10), we have

$$x_{13}^{(3)} > 0, \quad \Delta_3^{(3)} > 0.$$

We now express these parameters of B_3 in terms of the corresponding parameters of B. The basis B_3 is obtained from B by introducing the vector A_4 into the 2nd position. Hence, from the recurrence formulas (9.1) and (9.2)

$$x_{13}^{(3)}=x_{13}-\frac{x_{14}}{x_{24}}x_{23}>0, \quad \Delta_3^{(3)}=\Delta_3-\frac{\Delta_4}{x_{24}}x_{23}>0. \tag{9.22}$$

Thus, if chain (9.18) is possible, inequalities (9.19)-(9.22) are satisfied.

Let us now investigate this system. Applying the second inequality in (9.21), conditions (9.19), and the second inequality in (9.20), we obtain

$$x_{14}<0. \tag{9.23}$$

The second inequality in (9.20) can, therefore, be rewritten in the equivalent form

$$\Delta_3<\Delta_4\frac{x_{13}}{x_{14}}. \tag{9.24}$$

Comparing (9.24) with the second inequality in (9.22), we obtain

$$\Delta_4\frac{x_{23}}{x_{24}}<\Delta_4\frac{x_{13}}{x_{14}}.$$

But according to the second inequality in (9.21), $\Delta_4>0$. Hence

$$\frac{x_{23}}{x_{24}}<\frac{x_{13}}{x_{14}}. \tag{9.25}$$

Applying (9.23), we rewrite inequality (9.25) in the equivalent form

$$x_{13}<\frac{x_{14}x_{23}}{x_{24}}. \tag{9.26}$$

Inequality (9.26) contradicts the first inequality in (9.22). This proves the inconsistency of system (9.19)-(9.22). Chain (9.18) thus cannot arise.

9-5. Let us now increase the length of chain (9.18) by one. We obtain a new chain of bases:

$$B(A_1, A_2)\to B_1(A_3, A_2)\to B_2(A_3, A_4)\to B_3(A_5, A_4)\to B_4(A_5, A_2)\to B(A_1, A_2). \tag{9.27}$$

Let us write the conditions for transforming from B to B_1 and from B_1 to B_2, applying (9.7) and (9.8). Following the procedure in 9-4, we obtain inequalities (9.19) and (9.20).

The conditions of transformation from B_3 to B_4 and from B_4 to B_1 can be obtained from (9.9) and (9.10) as in the case of the chain (9.18). Omitting the details, we write out the corresponding inequalities:

$$x_{15}>0, \ \Delta_5>0, \tag{9.28}$$

$$\begin{gathered} x_{24}^{(4)}=x_{24}-\frac{x_{25}}{x_{15}}x_{14}>0, \\ \Delta_4^{(4)}=\Delta_4-\frac{\Delta_5}{x_{15}}x_{14}>0. \end{gathered} \tag{9.29}$$

It now remains to express the conditions for transformation from B_2 to B_3 in terms of the parameters of basis B. Applying (9.7) and (9.8) to the third step of chain (9.27), we obtain

$$x_{15}^{(2)}>0, \ \Delta_5^{(2)}<0.$$

Applying the recurrence formulas (9.3) and (9.4) for $i=r=1$, $r_1=2$, $j=5$, $k=3$, $k_1=4$, we obtain

$$x_{15}^{(2)}=\frac{x_{24}x_{15}-x_{14}x_{25}}{x_{24}x_{13}-x_{23}x_{14}}>0, \tag{9.30}$$

$$\Delta_5^{(2)} = \Delta_5 - \frac{x_{24}x_{15} - x_{14}x_{25}}{x_{24}x_{13} - x_{23}x_{14}}\Delta_3 - \frac{x_{25}x_{13} - x_{15}x_{23}}{x_{24}x_{13} - x_{23}x_{14}}\Delta_4 < 0. \tag{9.31}$$

Thus, cycle (9.27) may arise only if inequalities (9.19), (9.20) and (9.28)–(9.31) are consistent.

We shall show that these inequalities are inconsistent. From the first inequalities in (9.19) and (9.20) we obtain

$$x_{24}x_{13} - x_{23}x_{14} > 0. \tag{9.32}$$

Analogously, from the first inequalities in (9.28) and (9.29),

$$x_{24}x_{15} - x_{25}x_{14} > 0. \tag{9.33}$$

We shall distinguish between three possibilities depending on the sign of

$$\Delta = x_{25}x_{13} - x_{15}x_{23}. \tag{9.34}$$

1. Let $\Delta = 0$. Then, from (9.31), applying the second inequality in (9.19) and the inequalities (9.32) and (9.33), we obtain

$$\Delta_5 < 0.$$

On the other hand, from (9.28),

$$\Delta_5 > 0.$$

Hence, the first case is impossible.

2. Let now $\Delta > 0$. In this case, inequality (9.31) can be transformed to the equivalent form

$$\Delta_4 > \frac{x_{24}x_{13} - x_{23}x_{14}}{x_{25}x_{13} - x_{15}x_{23}}\Delta_5 - \frac{x_{24}x_{15} - x_{25}x_{14}}{x_{25}x_{13} - x_{15}x_{23}}\Delta_3. \tag{9.35}$$

From (9.35) and the second inequality in (9.20), we obtain

$$\Delta_3\frac{x_{14}}{x_{13}} > \Delta_4 > \frac{x_{24}x_{13} - x_{23}x_{14}}{x_{25}x_{13} - x_{15}x_{23}}\Delta_5 - \frac{x_{24}x_{15} - x_{25}x_{14}}{x_{25}x_{13} - x_{15}x_{23}}\Delta_3. \tag{9.36}$$

Applying inequalities (9.36), (9.32), (9.33), and the second inequality in (9.28), we obtain (for $\Delta > 0$)

$$\Delta_3\frac{x_{14}}{x_{13}} > \frac{x_{25}x_{14} - x_{15}x_{24}}{x_{25}x_{13} - x_{15}x_{23}}\Delta_3. \tag{9.37}$$

Since $\Delta_3 < 0$ (the second inequality in (9.19)), obvious transformations reduce (9.37) to

$$\frac{x_{15}(x_{14}x_{23} - x_{13}x_{24})}{x_{13}(x_{25}x_{13} - x_{15}x_{23})} > 0. \tag{9.38}$$

Hence, applying the first inequalities in (9.19), (9.28), and condition (9.32), we see that Δ is negative. The second case is also impossible.

3. It now remains to analyze the third possibility, namely $\Delta < 0$. In this case inequality (9.31) is equivalent to the condition

$$\Delta_4 < \frac{x_{24}x_{13} - x_{23}x_{14}}{x_{25}x_{13} - x_{15}x_{23}}\Delta_5 - \frac{x_{15}x_{24} - x_{25}x_{14}}{x_{25}x_{13} - x_{15}x_{23}}\Delta_3. \tag{9.39}$$

Since $\Delta_3 < 0$ (second inequality in (9.19)), $\Delta<0$, and inequality (9.33) applies, we have

$$\frac{x_{15}x_{24} - x_{25}x_{14}}{x_{25}x_{13} - x_{15}x_{23}}\Delta_3 > 0.$$

The second inequality in (9.29) and inequalities (9.39) thus lead to

$$\frac{x_{24}x_{13}-x_{23}x_{14}}{x_{25}x_{13}-x_{15}x_{23}}\Delta_5 > \Delta_4 > \frac{x_{14}}{x_{15}}\Delta_5. \tag{9.40}$$

From inequality (9.40) and second inequality in (9.28) it follows that

$$\frac{x_{13}(x_{15}x_{24}-x_{14}x_{25})}{x_{15}(x_{25}x_{13}-x_{15}x_{23})} > 0. \tag{9.41}$$

From (9.41), together with the first inequalities in (9.19), (9.28), and inequality (9.33), we have

$$\Delta = x_{25}x_{13} - x_{15}x_{23} > 0.$$

Once again we obtain a contradiction. The third possibility is also ruled out.

Thus we have proved the inconsistency of inequalities (9.19), (9.20), (9.28)–(9.31) associated with the sequence of bases (9.27). This indicates that the cycle (9.27) cannot arise. The preceding analysis culminates in the following general proposition:

Theorem 9.2. A cycle of length less than six iterations long cannot arise when solving by the simplex procedure.

Proof. Consider an arbitrary cycle of a linear-programming problem. Let new vectors be introduced into ζ basis positions in the course of this cycle. The length of the cycle, obviously, cannot be less than 2ζ. Therefore, if there exists a cycle whose length is at most 5, then $\zeta < \frac{5}{2}$. Consequently, we should consider only those cycles in which new vectors are introduced into one or two basis positions only. According to Theorem 9.1 no cycle in which new vectors are introduced into only one basis position can arise. Therefore, it remains to analyze the case $\zeta=2$. The minimum length of a cycle with $\zeta=2$ cannot be less than four steps. Hence, it suffices to study cycles of length 4 and 5 with $\zeta=2$.

If in two successive iterations of some cycle the new vectors are introduced into the same basis position, the length of the cycle can be reduced by one. Indeed, according to Lemma 9.1, the vector introduced into the basis in the second iteration is suitable to be introduced into the same basis position in the preceding iteration. Hence, the second iteration can be omitted, which reduces the cycle length by one. This shows that in the analysis of cycles of length 4 and 5 we may limit the discussion to chains of form (9.18) and (9.27), respectively. We have, however, proved that neither of these chains can arise. This completes the proof.

9-6. According to Theorem 9.2, any attempt to construct a cycle of less than six iterations a priori cannot succeed.

Consider a cycle of six bases of the form

$$\begin{aligned} &\mathrm{B}(A_1,\ A_2)\rightarrow \mathrm{B}_1(A_3,\ A_2)\rightarrow \mathrm{B}_2(A_3,\ A_4)\rightarrow \mathrm{B}_3(A_5,\ A_4)\rightarrow \\ &\qquad \rightarrow \mathrm{B}_4(A_5,\ A_6)\rightarrow \mathrm{B}_5(A_1,\ A_6)\rightarrow \mathrm{B}(A_1,\ A_2). \end{aligned} \tag{9.42}$$

In parentheses we give as always the vectors occupying the first two positions of the basis. The remaining basis positions remain unchanged ($\zeta=2$).

We write the conditions necessary and sufficient for chain (9.42) to arise. First, the first two basis variables x_{10} and x_{20} of the support program X corresponding to cycle (9.42) should be zero. The first three iterations of chain (9.42) coincide with the first three iterations of chain (9.27). The conditions of transformation from B to B_1, from B_1 to B_2, and from B_2 to B_3 therefore have the form (9.19), (9.20), and (9.30)–(9.31), respectively.

The conditions of transformation from B_6 to B, from B_4 to B_5, and from B_3 to B_4 are best obtained applying (9.9) and (9.10). Omitting the details, which do not differ from the corresponding transformations for chains (9.18) and (9.27), we give the final results:

$$x_{26} > 0, \quad \Delta_6 > 0 \tag{9.43}$$

(transformation from B_6 to B);

$$\begin{aligned} x_{15}^{(5)} &= x_{15} - \frac{x_{16}}{x_{26}} x_{25} > 0, \\ \Delta_5^{(5)} &= \Delta_5 - \frac{\Delta_6}{x_{26}} x_{25} > 0 \end{aligned} \tag{9.44}$$

(transformation from B_4 to B_5);

$$x_{24}^{(4)} = \frac{x_{15}x_{24} - x_{25}x_{14}}{x_{15}x_{26} - x_{16}x_{25}} > 0, \tag{9.45}$$

$$\Delta_4^{(4)} = \Delta_4 - \frac{x_{24}x_{15} - x_{14}x_{25}}{x_{15}x_{26} - x_{16}x_{25}} \Delta_6 - \frac{x_{14}x_{26} - x_{24}x_{16}}{x_{15}x_{26} - x_{16}x_{25}} \Delta_5 > 0 \tag{9.46}$$

(transformation from B_3 to B_4).

To derive the last two formulas, we applied equalities (9.3) and (9.4) for $i = r = 2$, $r_1 = 1$, $k = 6$, $k_1 = 5$, $j = 4$.

The first inequalities in (9.20), (9.44) and inequalities (9.30) and (9.45) are obviously equivalent to the conditions

$$\left.\begin{aligned} x_{15}x_{24} - x_{25}x_{14} > 0, \\ x_{15}x_{26} - x_{16}x_{25} > 0, \\ x_{24}x_{13} - x_{23}x_{14} > 0. \end{aligned}\right\} \tag{9.47}$$

The necessary and sufficient conditions for a cycle (9.42) to arise thus have the form

I.

$$x_{13} > 0, \tag{9.48}$$

$$x_{26} > 0, \tag{9.49}$$

$$x_{15}x_{24} - x_{25}x_{14} > 0, \tag{9.50}$$

$$x_{15}x_{26} - x_{16}x_{25} > 0, \tag{9.51}$$

$$x_{24}x_{13} - x_{23}x_{14} > 0. \tag{9.52}$$

II.

$$\Delta_3 < 0, \tag{9.53}$$

$$\Delta_6 > 0, \tag{9.54}$$

$$\Delta_4 - \frac{x_{14}}{x_{13}} \Delta_3 < 0, \tag{9.55}$$

$$\Delta_5 - \frac{x_{25}}{x_{26}} \Delta_6 > 0, \tag{9.56}$$

$$\Delta_5 - \frac{x_{24}x_{15} - x_{14}x_{25}}{x_{13}x_{24} - x_{14}x_{23}} \Delta_3 - \frac{x_{25}x_{13} - x_{15}x_{23}}{x_{13}x_{24} - x_{14}x_{23}} \Delta_4 < 0, \tag{9.57}$$

$$\Delta_4 - \frac{x_{24}x_{15} - x_{14}x_{25}}{x_{15}x_{26} - x_{16}x_{25}} \Delta_6 - \frac{x_{14}x_{26} - x_{24}x_{16}}{x_{15}x_{26} - x_{16}x_{25}} \Delta_5 > 0, \tag{9.58}$$

$$x_{10} = x_{20} = 0.$$

In system (9.48)–(9.58) the first five inequalities guarantee the positiveness of the corresponding coefficients in the expansion of the new vector in terms of the basis vectors. The last six inequalities in the system ensure negativeness of the evaluations of the new vectors with respect to the current basis.

We shall prove consistency of system (9.48)–(9.58), isolating a certain class of its solutions. Let

$$\Delta_4 = \Delta_6 = -\Delta_3 = -\Delta_5 = a > 0, \tag{9.59}$$

$$\left.\begin{aligned} x_{14}=x_{25}=\alpha,\\ x_{13}=x_{26}=\beta,\\ x_{24}=x_{15}=\gamma,\\ x_{23}=x_{16}=\delta. \end{aligned}\right\} \tag{9.60}$$

Under these conditions, inequalities (9.53), (9.54) are automatically satisfied. Inequalities (9.48), (9.49) are equivalent to the requirement

$$\beta>0. \tag{9.61}$$

Inequalities (9.50)–(9.52) are equivalent to

$$\gamma^2-\alpha^2>0, \tag{9.62}$$
$$\gamma\beta-\alpha\delta>0. \tag{9.63}$$

Applying (9.59), we can easily show that conditions (9.55)–(9.58) are, respectively, equivalent to the inequalities

$$\alpha+\beta<0, \tag{9.64}$$
$$(\gamma+\alpha)(\beta+\alpha-\gamma-\delta)>0. \tag{9.65}$$

System (9.48)–(9.58), subject to the additional assumptions (9.59), (9.60), is, thus, equivalent to system (9.61)–(9.65). Let us consider this system, comprising five inequalities.

1. Let $\gamma>0$. Then, from (9.62),

$$(\gamma+\alpha)>0. \tag{9.66}$$

Applying (9.66) and (9.65), we obtain

$$\delta<\beta+\alpha-\gamma. \tag{9.67}$$

From (9.61) and (9.64), $\alpha<0$. Hence, (9.63) can be rewritten in the form

$$\delta>\frac{\gamma\beta}{\alpha}. \tag{9.68}$$

Comparing (9.67) and (9.68), we have

$$-\frac{\gamma\beta}{|\alpha|}<-|\beta+\alpha|-\gamma. \tag{9.69}$$

On the other hand, from (9.64), $\frac{\beta}{|\alpha|}<1$. Therefore,

$$-\gamma<-\frac{\gamma\beta}{|\alpha|}. \tag{9.70}$$

Comparison of (9.69) and (9.70) gives the impossible relationship

$$|\beta+\alpha|<0.$$

Hence, if $(\alpha,\ \beta,\ \gamma,\ \delta)$ is a solution of system (9.61)–(9.65) $\gamma>0$(for $\gamma=0$ inequality (9.62) is not satisfied).

2. Let $\gamma<0$. Then inequalities (9.63), (9.65) are, respectively, rewritten in the form

$$\delta>\frac{|\gamma|}{|\alpha|}\beta, \quad \delta>|\gamma|-|\beta+\alpha|.$$

Therefore, for $\delta>|\gamma|$ both these inequalities are satisfied (according to (9.64), $\frac{\beta}{|\alpha|}<1$). The other three inequalities are equivalent to

$$0<\beta<-\alpha<-\gamma.$$

2154

Therefore, any set $(\alpha, \beta, \gamma, \delta)$ satisfying the conditions

$$0 < \beta < -\alpha < -\gamma < \delta, \tag{9.71}$$

is a solution of inequalities (9.61)–(9.65).

Let us now choose arbitrary α, β, γ, δ, satisfying (9.71).

Applying formulas (9.60) we then determine the parameters x_{ij}. Next we choose linear-form coefficients c_j such that the evaluations of the corresponding restraint vectors satisfy inequalities (9.59). Then the parameters x_{ij} and $\Delta_j (i = 1, 2;\ j = 3, 4, 5, 6)$ constitute a solution of inequalities (9.58)–(9.68) corresponding to cycle (9.42). We have thus obtained an entire class of linear-programming problems in each of which a cycle of six bases may arise.

This result and Theorem 9.2 lead to the following proposition.

Theorem 9.3. The minimum length of a cycle in linear-programming problems is six iterations.

9-7. We now give an example illustrating the results of 9-6.

Consider the following linear-programming problem with seven nonnegative variables having three equality restraints.

Maximize the linear form

$$L(X) = x_3 - x_4 + x_5 - x_6$$

subject to the conditions

$$\begin{aligned} x_1 \quad + x_3 - 2x_4 - 3x_5 + 4x_6 \quad &= 0, \\ x_2 + 4x_3 - 3x_4 - 2x_5 + x_6 \quad &= 0, \\ x_3 + x_4 + x_5 + x_6 + x_7 &= 1. \end{aligned}$$

The compilation of the initial tableau for this problem is obvious (Table 5.20). We see from this tableau

TABLE 5.20

No.	C_X	B	A_0 \ c_j	A_1	A_2	A_3 (1)	A_4 (−1)	A_5 (1)	A_6 (−1)	A_7	θ	No. of tableau
1		A_1		1		1	−2	−3	4			
2		A_2			1	4	−3	−2	1			0
3		A_7	1			1	1	1	1	1	1	
4	—	—				−1	1	−1	1			

that the problem in question belongs to the class of problems constructed in 9-6. Here

$$\begin{aligned} \alpha &= x_{14} = x_{25} = -2, \\ \beta &= x_{13} = x_{26} = 1, \\ \gamma &= x_{24} = x_{15} = -3, \\ \delta &= x_{23} = x_{16} = 4, \\ x_{10} &= x_{20} = 0, \\ a = -\Delta_3 &= \Delta_4 = -\Delta_5 = \Delta_6 = 1. \end{aligned}$$

Conditions (9.71) are obviously satisfied. Therefore, when solving this problem by the simplex method, we may encounter cycling. We give here a sequence of tableaus (see Table 5.21) corresponding to the cycle in question.

In the first step the vector $A_3 (\Delta_3 = -1)$ is introduced into the basis. Here $\theta_0 = 0$ is obtained on the first two positions of the basis. The restraint vector A_3 is placed in the first vector position. Further, the vector $A_4 (\Delta'_4 = -1)$ is introduced into the basis. Since $\theta_0 = 0$ is obtained only on one basis position (the 2nd),

TABLE 5.21 (1-6)

	No.	c_X	B	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	θ	No. of tab-leau
→	1	1	A_3		1		1	−2	−3	4		—	1
←	2		A_2		−4	1		5	10	−15			
	3		A_7	1	−1			3	4	−3	1	$1/3$	
	4	—	—		1			−1	−4	5			
←	1	1	A_3		$-3/5$	$2/5$	1		1	−2			2
→	2	−1	A_4		$-4/5$	$1/5$		1	2	−3			
	3		A_7	1	$7/5$	$-3/5$			−2	6	1	—	
	4	—	—		$1/5$	$1/5$			−2	2			
→	1	1	A_5		$-3/5$	$2/5$	1		1	−2		—	3
←	2	−1	A_4		$2/5$	$-3/5$	−2	1		1			
	3		A_7	1	$1/5$	$1/5$	2			2	1	$1/2$	
	4	—	—		−1	1	2			−2			
←	1	1	A_5		$1/5$	$-4/5$	−3	2	1				4
→	2	−1	A_6		$2/5$	$-3/5$	−2	1		1			
	3		A_7	1	$-3/5$	$7/5$	6	−2			1	—	
	4	—	—		$-1/5$	$-1/5$	−2	2					
→	1		A_1		1	−4	−15	10	5			—	5
←	2	−1	A_6			1	4	−3	−2	1			
	3		A_7	1		−1	−3	4	3		1	—	
	4	—	—			−1	−5	4	1				
	1		A_1		1		1	−2	−3	4			6
→	2		A_2			1	4	−3	−2	1			
	3		A_7	1			1	1	1	1	1		
	4	—	—				−1	1	−1	1			

the vector A_4 is placed in the 2nd position. In subsequent iterations, the vectors A_5, A_6, A_1, A_2 are introduced in the 1st, 2nd, 1st, 2nd basis positions, respectively.

We see from Table 5.21 that the basis position in which a new vector was to be introduced in the 1st, 3rd, and 5th iterations was not chosen in any specific manner. In these cases the position was chosen arbitrarily. It is obvious that had we used the exact rule described in 2-3 for choosing the position, no cycling would have arisen.

For example, let us apply the exact rule in the first iteration:

$$\frac{x_{10}}{x_{13}}=\frac{x_{20}}{x_{23}}=\theta_0=0.$$

Following this rule, we compare the ratios

$$\frac{x_{11}}{x_{13}}=1 \text{ and } \frac{x_{21}}{x_{23}}=0.$$

Since $\frac{x_{11}}{x_{13}}>\frac{x_{21}}{x_{23}}$, the vector A_3 should be introduced into the second basis position. The reader will verify that, having applied this rule, the linear form is increased already in the next iteration.

9-8. The results of the present section show that an arbitrary choice of the position in which a new vector is introduced into the basis may produce cycling. We have, however, seen that for cycling to occur the problem parameters should satisfy fairly rigid requirements. As expected cycling is, therefore, a rare phenomenon.

Indeed, in none of the linear-programming problems solved in the applied examples has a cycle arisen although most of them have been solved without using the exact rule for elimination of one of the basis vectors. The first published example of a cycle is due to Beale /6/. In the literature there are also some references to an unpublished example of a cycle constructed earlier by Hoffman /36/.

We thus reach the following conconlusions.

1. Although in principle cycling may occur, applied linear-programming problems should be solved with the aid of the simplified rule for choosing the vector to be eliminated from the basis. One of these rules is given in § 2.

2. In theoretical applications of the simplex method, to avoid cycling the exact rules developed in Chapter 4, § 6, must be applied. Otherwise, the class of problems for which any results obtained will be valid is a priori limited.

EXERCISES TO CHAPTER 5

1. Consider a two-parameter (a and b) family of linear-programming problems.
Maximize the linear form

$$L(X)=\sum_{j=1}^{6} x_j$$

subject to the conditions

$$\begin{gathered} x_1+2x_2-x_3+ax_4+bx_5+x_6=2, \\ x_1-3x_2+4x_3+bx_4+ax_5+bx_6=7, \\ x_1+5x_2-2x_3+x_4+bx_5+ax_6=5, \\ x_j\geqslant 0, \quad j=1, 2, \ldots, 6. \end{gathered}$$

Indicate the domain of definition of the parameters a and b corresponding to case (a) (optimal program), case (b) (unsolvable problem), and case (c) (improvable program) with reference to the following support program of the problem:

$$X=(1, 2, 3, 0, 0, 0).$$

Give a graphical representation of the corresponding domains in the (a, b)-plane.

2. Applying the first simplex algorithm, maximize the linear form

$$L(X)=x_1+2x_2-x_3+4x_4-3x_5$$

subject to the conditions

$$\begin{aligned} x_1-x_2+3x_3-x_4+2x_5&\leq 10,\\ 2x_2+3x_4-x_5&\leq 6,\\ 3x_1+4x_2+x_3+2x_4+x_5&\leq 25,\end{aligned}$$

$$x_j\geqslant 0,\quad j=1,\ldots,5.$$

3. Solve Exercise 2 applying the second simplex algorithm.

4. Apply the product form of the second algorithm to the solution of Exercise 2.

5. Solve Exercise 2 applying the comments of 5-3 on the determination of the vector to be introduced into the basis.

6. Draw a block diagram of the product form of the second algorithm.

7. Solve Exercise 2 with additional restraints

$$x_j\leq 1,\quad j=1,2,\ldots,5,$$

considering the problem as a problem with bilateral restraints.

8. Determine a support program of a linear-programming problem with the following system of restraints:

$$\begin{aligned} 3x_1-2x_2+x_3-3x_4+4x_5&=2,\\ 5x_1+3x_2-2x_3+x_4+x_5&=6,\\ x_1-2x_2+4x_3-2x_4&=3,\end{aligned}$$

$$x_j\geqslant 0,\quad j=1,\ldots,5.$$

9. Determine a support program of a problem with the following system of restraints:

$$\begin{aligned} 5x_1+2x_2-x_3+3x_4-x_5+2x_6&=6,\\ 2x_1-3x_2+2x_3+x_4+2x_5+x_6&=1,\\ 16x_1-5x_2+4x_3+9x_4+4x_5+7x_6&=15,\end{aligned}$$

$$x_j\geqslant 0,\quad j=1,2,\ldots,6.$$

Write the maximum number of linearly independent equations of the system and write the equivalent problem.

Hint: Use the comments of 8-1.

10. Maximize the linear form

$$L(X)=x_1+x_2-3x_3+x_4+x_5$$

subject to the restraints of Exercise 8:

(a) applying the support program of Exercise 8 as the initial program;

(b) applying the M-method.

11. Prove formulas (9.3) and (9.4).

12. Consider the following linear-programming problem:

Maximize the linear form

$$L(X)=x_3-x_4+x_5-x_6$$

subject to the conditions

$$\begin{aligned} x_1+2x_3-3x_4-5x_5+6x_6&=0,\\ x_2+6x_3-5x_4-3x_5+2x_6&=0,\\ 3x_3+x_4+2x_5+4x_6+x_7&=1,\end{aligned}$$

$$x_j\geqslant 0,\quad j=1,2,\ldots,7.$$

Taking the vectors (A_1, A_2, A_7) as the initial basis, show that a cycle may arise if solution of the problem is attempted without applying the rule which guarantees agains cycling.

Chapter 6

THE DUAL SIMPLEX METHOD

Several methods suitable for solving linear-programming problems follow from the duality theory discussed in Chapter 3. From a synthesis of the simplex method and the principal concepts of duality we have the so-called *dual simplex method*, which was first described by Lemke /73/ in 1954.

The solution of a linear-programming problem by the dual simplex method (which is sometimes simply called the *dual method*) involves determining an optimal program of the dual problem and hence, by the duality theorems, an optimal program of the primal problem, too.

In Chapter 3, 5-5, we showed that the components of the optimal program of the dual problem give an estimation of the influence of the various restraints of the linear-programming problem on the magnitude of the maximum of the linear form. The method presented in this chapter, proceeding from approximate (preliminary) evaluations of restraints of the primal problem (from an initial program of the dual problem), yields, after successive refinements of these evaluations, a vector of exact evaluations of the problem restraints (an optimal program of the dual problem). The system of preliminary restraint evaluations obtained in each iteration of the dual method can be made to correspond to an n-dimensional vector X which satisfies the equality restraints of the primal problem, but may have negative components. The system of restraint evaluations obtained in the last iteration corresponds to a vector X^* with nonnegative components. The optimal support program is, thus, obtained as the result of successive refinement of restraint evaluations; this is reflected in the name given to the method in Soviet literature — *the method of successive evaluation refinement*.

We note that when passing from a given iteration to the next the value of the linear form decreases monotonically. Thus, unlike the simplex method where the maximum of the linear form is approached from below, the dual method leads to the optimum from above.

The dual simplex method is presented as follows. First, in § 1, we state the optimality test and discuss the theoretical principles of the dual method with reference to the nondegenerate case. In § 2 the formal description of the method is supplemented by two geometrical interpretations. § 3 deals with applications of the dual simplex method to linear-programming problems with bilaterally restrained variables.

In the first three sections the dual problem is assumed to be nondegenerate. In § 4 we establish rules which can be followed when applying the dual method to degenerate problems; these rules guarantee against cycling in the degenerate case.

In § 5 and § 6 we discuss two computational procedures, two algorithms of the dual simplex method. Each of these dual algorithms proceeds from a given initial support program of the dual problem. In § 7 we consider various methods for determining the initial support program of the dual problem. In the last section, § 8, the simplex and the dual simplex methods are compared and the problem of applying them simultaneously is considered.

§ 1. Principles

1-1. We write the linear-programming problem in canonical form. Maximize the linear form

$$L(X)=\sum_{j=1}^{n}c_jx_j \tag{1.1}$$

subject to the conditions

$$\sum_{j=1}^{n}a_{ij}x_j=b_i,\qquad i=1,2,\ldots,m; \tag{1.2}$$

$$x_j\geqslant 0,\qquad j=1,2\ldots,n. \tag{1.3}$$

As always we assume the matrix $\|a_{ij}\|_{m,n}$ of restraints (1.2) to be of rank m. The dual problem of problem (1.1)–(1.3) involves minimizing the linear form

$$\tilde{L}(Y)=\sum_{i=1}^{m}b_iy_i \tag{1.4}$$

subject to the conditions

$$\sum_{i=1}^{m}a_{ij}y_i\geqslant c_j,\qquad j=1,2,\ldots,n. \tag{1.5}$$

The primal and the dual problems can be written more compactly in the following form:

$L(X)=(C,X)\to\max$ subject to the conditions $AX^T=B,\ X\geqslant 0;$

$\tilde{L}(Y)=(B,Y)\to\min$ subject to the conditions $YA\geqslant C.$

Let us now define some concepts associated with the dual problem.

A program Y of the dual problem is said to be a *support* program if among the restraints (1.5) reduced to equalities on this program there exist m linearly independent restraints (see Chapter 2, § 5).

The *basis* of a support program Y of the dual problem (1.4)–(1.5) is any system of m linearly independent restraint vectors A_{s_i} of the primal problem such that

$$(Y,A_{s_\mu})=c_{s_\mu},\qquad \mu=1,2,\ldots,m,$$

or, equivalently,

$$\sum_{i=1}^{m}a_{is_\mu}y_i=c_{s_\mu},\qquad \mu=1,2,\ldots,m. \tag{1.6}$$

Let I_Y denote the set of indices $(s_1,s_2,\ldots,s_m)$ of the basis of the support program Y. The system of vectors $A_{s_1},\ldots,A_{s_m}$ comprising the basis of some support program of the dual problem is called, for brevity, the *dual basis*.

A support program Y of the dual problem is said to be nondegenerate if for any vector A_j not appearing in the basis

$$(Y, A_j)>c_j \tag{1.7}$$

(see Chapter 2, § 5).

The dual problem (1.4)–(1.5) whose support programs are all nondegenerate is called a nondegenerate problem. Geometrically, nondegeneracy indicates that exactly m boundary hyperplanes intersect at each vertex of the polyhedron (polyhedral set).

In the following sections the dual problem is assumed to be nondegenerate.

Each support program Y of the dual problem (more precisely, each support-program basis) is conveniently made to correspond to some n-dimensional vector X satisfying restraints (1.2) of the primal problem. This correspondence will enable us, in the following, to translate all the constructions pertaining to support programs of the dual problem into terms of the primal problem.

Let us expand the constraint vector B in terms of the dual basis $(A_{s_1}, \ldots, A_{s_m})$. Let x_{s_i} be the corresponding expansion coefficients. The n-dimensional vector X whose s_i-th components ($i=1, 2, \ldots, m$) are identical with x_{s_i} and whose other components are all zero will be called a pseudoprogram of the primal program. All x_{s_i} will be called basis components of the pseudoprogram, and the others extrabasis variables. The vectors of the dual basis, like those of the primal basis, can be characterized not only by their indices but also by their respective positions in the basis.

The basis component x_{s_i} of the pseudoprogram will be denoted by x_{i0} according to the position i of the vector A_{s_i} in the dual basis. The pseudoprogram components obviously satisfy restraints (1.2) of the primal problem:

$$\sum_{j=1}^{n} a_{ij}x_j=\sum_{\mu=1}^{m} a_{is_\mu}x_{s_\mu}=b_i, \qquad i=1, 2, \ldots, m. \tag{1.8}$$

It must be kept in mind that some x_{s_μ} may be negative, since X, generally speaking, is not a feasible program of the primal problem. In what follows the terms dual basis and pseudoprogram basis will be used synonymously.

A pseudoprogram can also be defined independently of the dual problem. Let $S=(A_{s_1}, \ldots, A_{s_m})$ be a system of linearly independent restraint vectors. Let I be the system of indices $s_1, \ldots, s_m$. We expand the restraint vectors A_j ($j=1, 2, \ldots, n$) and the constraint vector $B=A_0$ in terms of the vectors of systems S:

$$A_j=\sum_{i=1}^{m} x_{ij}A_{s_i}, \tag{1.9}$$

$$B=A_0=\sum_{i=1}^{m} x_{i0}A_{s_i}. \tag{1.10}$$

We take

$$\Delta_j=\sum_{i=1}^{m} c_{s_i}x_{ij}-c_j, \qquad j=1, 2, \ldots, n. \tag{1.11}$$

Theorem 6.1. *An n-dimensional vector X such that $x_{s_i}=x_{i0}$ and $x_j=0$ for $j\notin I$ is a pseudoprogram if and only if all $\Delta_j\geqslant 0$ ($j=1, 2, \ldots, n$).*

Proof. The vectors $A_{s_1}, \ldots, A_{s_m}$ are linearly independent. We may, therefore, compute the vector $Y=(y_1, \ldots, y_m)$ whose components satisfy the

conditions

$$\sum_{\mu=1}^{m} a_{\mu s_i} y_\mu = c_{s_i}.$$

We have

$$\Delta_j = \sum_{i=1}^{m} c_{s_i} x_{ij} - c_j = \sum_{i=1}^{m} \left(\sum_{\mu=1}^{m} a_{\mu s_i} y_\mu\right) x_{ij} - c_j = \sum_{\mu=1}^{m} y_\mu \left(\sum_{i=1}^{m} a_{\mu s_i} x_{ij}\right) - c_j.$$

It follows from (1.9) that

$$a_{\mu j} = \sum_{i=1}^{m} a_{\mu s_i} x_{ij}, \tag{1.12}$$

therefore

$$\Delta_j = \sum_{i=1}^{m} c_{s_i} x_{ij} - c_j = \sum_{\mu=1}^{m} a_{\mu j} y_\mu - c_j. \tag{1.13}$$

This relationship shows the equivalence of the two definitions of a pseudoprogram. This completes the proof.

We must emphasize that while a feasible program of a problem is specified by the problem restraints only, a pseudoprogram is defined by both the restraints and the linear form of the problem.

The dual simplex method involves application of the simplex method to the dual problem. This approach gives a new technique for solving the primal problem.

Consider a support program $Y=(y_1, \ldots, y_m)$ of the dual problem. The basis of support program Y comprises the vectors $A_{s_1}, \ldots, A_{s_m}$. This corresponds to pseudoprogram X with the basis components x_{s_i} $(i=1, 2, \ldots, m)$.

Optimality test. *If all the basis components of pseudoprogram X are nonnegative, pseudoprogram X is an optimal program of the primal problem, and support program Y is a solution of the dual problem.*

Proof. We have the following chain of equalities:

$$\tilde{L}(Y) = \sum_{i=1}^{m} b_i y_i = \sum_{i=1}^{m} \left(\sum_{\mu=1}^{m} a_{is_\mu} x_{s_\mu}\right) y_i = \sum_{\mu=1}^{m} x_{s_\mu} \left(\sum_{i=1}^{m} a_{is_\mu} y_i\right) = \sum_{\mu=1}^{m} c_{s_\mu} x_{s_\mu} = \sum_{j=1}^{n} c_j x_j = L(X).$$

Here the second equality follows from (1.8) for pseudoprogram components. The third equality is obtained by reversing the order of summation. The fourth relationship follows from (1.6). The fifth equality holds because all the extrabasis components of a pseudoprogram are zero.

With nonnegative x_j, pseudoprogram X is a feasible program of the primal problem. According to Chapter 3, Lemma 1.2, the programs of a dual pair related by the expression

$$\tilde{L}(Y) = L(X),$$

are optimal programs of the corresponding problems. This completes the proof.

Observe that the above optimality test follows directly from the second form of the simplex optimality test, since the requirements of the second form of the test are satisfied for pseudoprogram X. In the general case, the above test is only sufficient to establish program optimality. We leave it to the reader to prove that this test is also necessary, if the program Y of the dual problem is nondegenerate (see Exercise 1).

1-2. Let a support program Y of the dual problem with the basis $A_{s_1}, \ldots, A_{s_m}$ be given. This corresponds to pseudoprogram X of the primal problem. The components in the expansion of the constraint vector $B=A_0$

and of the restraint vectors A_j of the primal problem in terms of the vectors of the dual basis can be computed from the equalities

$$A_j = \sum_{i=1}^{m} x_{ij} A_{s_i}, \qquad j = 0, 1, 2, \ldots, n. \tag{1.14}$$

The parameter x_{ij} in (1.14) is the coefficient in the expansion of the restraint vector A_j in terms of the vectors of the dual basis associated with the vector A_{s_i} occupying the i-th position in the dual basis.

In what follows, we shall require an alternative technique for computing the coefficients x_{ij}:

$$x_{ij} = \sum_{\mu=1}^{m} a_{\mu j} e_{i\mu}, \qquad i = 1, 2, \ldots, m; \quad j = 0, 1, 2, \ldots, n, \tag{1.15}$$

where $e_{i\mu}$ are the coefficients in the expansion of the m-dimensional unit vectors e_μ ($\mu = 1, 2, \ldots, m$) in terms of the vectors of the dual basis, i.e.,

$$\| e_{ij} \|_m = (A_{s_1}, A_{s_2}, \ldots, A_{s_m})^{-1}. \tag{1.16}$$

Formula (1.15) follows from the matrix relationships

$$A_j = A_Y X_j$$

or

$$X_j = A_Y^{-1} A_j,$$

where

$$X_j = (x_{1j}, x_{2j}, \ldots, x_{mj})^T, \quad A_Y = (A_{s_1}, A_{s_2}, \ldots, A_{s_m}).$$

We distinguish between three cases depending on the signs of the basis components of the pseudoprogram and of the coefficients in the expansion of the restraint vectors A_j in terms of the dual basis:

a) The basis components x_{i0} are nonnegative for all i ($i = 1, 2, \ldots, m$).

b) Some of the x_{i0} are negative and for at least one of them all $x_{ij} \geqslant 0$ for $j = 1, 2, \ldots, n$.

c) The pseudoprogram has negative basis components x_{i0}, but each has some negative coefficients x_{ij} ($j = 1, 2, \ldots, n$).

In case (a), according to the optimality test, the pseudoprogram is an optimal support program of the primal problem. In what follows we shall show that in case (b) the linear-programming problem is unsolvable (the restraints of the primal problem are inconsistent), and in case (c) we may advance to a new support program of the dual problem and, consequently, to a new pseudoprogram with a lower value of the linear form.

1-3. To simplify the analysis of cases (b) and (c), it is advisable to follow the variation of linear form (1.4) under transformation from support program Y to a new program

$$Y(\theta) = Y + \theta e^{(i)}. \tag{1.17}$$

Here $e^{(i)} = (e_{i1}, \ldots, e_{im})$; e_{ij} are the coefficients in the expansion of the unit vectors

$$e_j = (\overbrace{\underbrace{0, \ldots, 0, 1}_{i}, 0, \ldots, 0}^{m})$$

in terms of the basis vectors of program Y. To simplify notations in

relationships containing the vector $e^{(i)}$, the number of the basis to which the components of this vector correspond is not indicated.

The transition from program Y to program $Y(\theta)$ will be called the elementary transformation of program Y associated with the vector A_{s_i} of the dual basis.

Inserting the components of the vector $Y(\theta)=\{y_1(\theta), \ldots, y_m(\theta)\}$ into the left-hand side of restraints (1.5) of the dual problem, we obtain

$$\sum_{\mu=1}^{m} a_{\mu j} y_{\mu}(\theta)=\sum_{i=1}^{m} a_{\mu j} y_{\mu}+\theta \sum_{\mu=1}^{m} a_{\mu j} e_{i\mu}.$$

From (1.15)

$$\sum_{\mu=1}^{m} a_{\mu j} e_{i\mu}=x_{ij}, \qquad j=0,\ 1,\ \ldots,\ n.$$

In particular, for $j \in I_Y=(s_1,\ s_2,\ \ldots,\ s_m)$

$$\sum_{i=1}^{m} a_{\mu j} e_{i\mu}=\begin{cases} 0 \text{ for } j \in I_Y,\ j \neq s_i, \\ 1 \text{ for } j=s_i. \end{cases}$$

We have from these formulas

$$\sum_{\mu=1}^{m} a_{\mu j} y_{\mu}(\theta)=\begin{cases} \sum\limits_{\mu=1}^{m} a_{\mu j} y_{\mu}, & j \in I_Y,\ i \neq s_i, \\ \sum\limits_{\mu=1}^{m} a_{\mu j} y_{\mu}+\theta, & j=s_i, \\ \sum\limits_{\mu=1}^{m} a_{\mu j} y_{\mu}+\theta x_{ij}, & j \notin I_Y. \end{cases} \tag{1.18}$$

We have previously defined in (1.11)

$$\Delta_j=\sum_{\mu=1}^{m} a_{\mu j} y_{\mu}-c_j, \qquad j=1,\ 2,\ \ldots,\ n.$$

Since $Y=(y_1,\ \ldots,\ y_m)$ is a program of the dual problem, we have

$$\Delta_j \geqslant 0 \text{ for } j=1,\ 2,\ \ldots,\ n. \tag{1.19}$$

The vector $Y(\theta)$ is a feasible program of the dual problem if and only if

$$\Delta_j(\theta)=\sum_{\mu=1}^{m} a_{\mu j} y_{\mu}(\theta)-c_j \geqslant 0, \qquad j=1,\ 2,\ \ldots,\ n. \tag{1.20}$$

Let us now determine the conditions under which all (1.20) are satisfied. Applying (1.11), (1.18), and (1.20), we obtain

$$\Delta_j(\theta)=\begin{cases} \Delta_j=0, & \text{if } j \in I_Y,\ j \neq s_i, \\ \Delta_j+\theta=\theta, & \text{if } j=s_i, \\ \Delta_j+\theta x_{ij}, & \text{if } j \notin I_Y. \end{cases} \tag{1.21}$$

It follows from (1.21) for $j=s_i$ that $\theta \geqslant 0$. If $x_{ij} \geqslant 0$, then $\Delta_j(\theta) \geqslant \Delta_j \geqslant 0$. If $x_{ij}<0$, conditions (1.20) are satisfied if

$$\Delta_j(\theta)=\Delta_j+\theta x_{ij} \geqslant 0,$$

i. e.,

$$\theta \leqslant -\frac{\Delta_j}{x_{ij}}.$$

$Y(\theta)$ is thus a feasible program of the dual problem (1.4)–(1.5) for all θ in the interval

$$0 \leqslant \theta \leqslant \theta_0, \tag{1.22}$$

where

$$\theta_0 = \min_{x_{ij}<0} \left(-\frac{\Delta_j}{x_{ij}}\right). \tag{1.23}$$

If $x_{ij} \geqslant 0$ for $j=1, 2, \ldots, n$, θ is bounded above, i.e., $\theta_0 = \infty$. For a nondegenerate program all $\Delta_j > 0$, $j \neq I_Y$, so that $\theta_0 > 0$.

The value of the linear form $\tilde{L}(Y)$ of the dual problem on program $Y(\theta)$ is

$$\tilde{L}[Y(\theta)] = \sum_{\mu=1}^{m} b_\mu y_\mu(\theta) = \sum_{\mu=1}^{m} b_\mu (y_\mu + \theta e_{i\mu}).$$

According to (1.15), for $j=0$ we have

$$x_{i0} = \sum_{\mu=1}^{m} a_{\mu 0} e_{i\mu} = \sum_{\mu=1}^{m} b_\mu e_{i\mu},$$

so that

$$\tilde{L}[Y(\theta)] = \tilde{L}(Y) + \theta x_{i0}. \tag{1.24}$$

1-4. We will now analyze cases (b) and (c) outlined in 1-2.

Let case (b) obtain: among the negative components x_{i0} of pseudoprogram X there is a component x_{r0} for which $x_{rj} \geqslant 0$, $j=1, 2, \ldots, n$. In this case $Y(\theta)$—the result of the elementary transformation (associated with the vector A_{s_r}) of program Y—is a feasible program of the dual problem for any θ. Hence, we see from (1.24) (for $i=r$) that linear form $\tilde{L}(Y)$ is unbounded below in the set of feasible programs of the dual problem. According to Lemma 1.3, Chapter 3, this is possible only if the restraints of the primal problem are inconsistent.

Note that in the simplex method unsolvability of a problem resulted from unboundedness of the linear form in the set of feasible programs. The case of inconsistent restraints did not occur, since in advance we assumed the existence of an initial program of the primal problem. When analyzing the dual simplex method, we start with an initial program of the dual problem. This ensures boundedness of the linear form of the primal problem, but does not guarantee consistency of its restraints.

Consider case (c). Let the pseudoprogram have negative basis components such that each has some negative coefficients x_{ij}. In this case the vector $Y(\theta)$ is a feasible program of the dual problem for any $\theta \geqslant 0$ not exceeding θ_0 in (1.23). Take a negative basis component $x_{r0} < 0$ of pseudoprogram X. By assumption, some of the x_{rj} are negative. Let the index j on which θ_0 is attained be k:

$$\theta_0 = -\frac{\Delta_k}{x_{rk}}.$$

Obviously,

$$\Delta_k(\theta_0) = \Delta_k + \theta_0 x_{rk} = 0.$$

According to Chapter 4, Theorem 2.1, the system of vectors obtained from the dual basis of program Y when the vector A_{s_r} is replaced by A_k is linearly independent, since by assumption $x_{rk} \neq 0$ ($x_{rk} < 0$). Hence $Y' = Y(\theta_0)$ is not just a feasible program, but is actually a support program of the dual problem. The basis of the support program Y' is obtained from $A_{s_1}, \ldots, A_{s_m}$ when A_{s_r} is substituted for A_k.

By assumption, the problem in question is nondegenerate, so that Y' is a nondegenerate program, i.e., $\Delta_j(\theta_0) > 0$ for $j \notin I_{Y'}$. Hence, θ_0 is obtained on a single vector A_k. In the nondegenerate case the vector to be introduced into the basis is thus uniquely determined.

We observe from (1.24) that when passing from program Y to $Y' = Y(\theta_0)$, the linear form of the dual problem decreases by

$$|\theta_0 x_{r0}| > 0 \quad (\theta_0 > 0,\ x_{r0} < 0).$$

Observe that elementary transformation of support program Y into support program $Y' = Y(\theta_0)$ corresponds to a transition from pseudoprogram X to pseudoprogram X' of the primal problem.

The value of the linear form $L(X)$ of the primal problem on the pseudoprogram is equal to the value of the linear form $\tilde{L}(Y)$ on the corresponding support program of the dual problem.

Indeed,

$$L(X) = \sum_{j=1}^{n} c_j x_j = \sum_{j \in I_Y} c_j x_j = \sum_{i=1}^{m} c_{s_i} x_{i0} = \sum_{i=1}^{m} x_{i0} \sum_{\mu=1}^{m} a_{\mu s_i} y_\mu =$$
$$= \sum_{\mu=1}^{m} y_\mu \sum_{i=1}^{m} a_{\mu s_i} x_{i0} = \sum_{\mu=1}^{m} y_\mu b_\mu = \tilde{L}(Y). \qquad (1.25)$$

Thus, in case (c) elementary transformation of the support program Y of the dual problem associated with the vector A_{s_r} decreases the linear form $\tilde{L}(Y)$. In terms of the primal problem this means that in case (c) we may pass from pseudoprogram X to a pseudoprogram X' with a lower value of the linear form $L(X)$.

Successive transformations from a given support program Y of the dual problem to another (or, in terms of the primal problem, successive transformations of one pseudoprogram to another) are continued until a solution of the problem is obtained or unsolvability is established.

Each transformation from a given pseudoprogram to the successive one constitutes an *iteration (step)* of the dual simplex method. When this method is applied to the solution of problems with a nondegenerate dual problem, we may not return to a basis which has once been examined. Otherwise, the support programs of the dual problem obtained in different iterations would be coincident. This, however, is impossible since the linear form of the dual problem decreases monotonically. The number of iterations required to reach a solution of a nondegenerate problem (or to establish its unsolvability) is a priori bounded by the total number of bases of the dual problem, which does not exceed C_n^m.

1-5. We now briefly outline the sequence of operations involved in a single iteration of the dual method. Each iteration comprises two stages. In the first stage we check whether the pseudoprogram is a feasible program of the primal problem and, if not, whether the problem is solvable. To this end we compute the basis components of the pseudoprogram (expand the constraint vector in terms of the vectors of the dual basis) and establish their signs.

The pseudoprogram proves to be a feasible program and, consequently, a solution of the problem (case (a)) if all its components are nonnegative. If the optimality test is not satisfied, we must verify that the restraints of the primal problem are consistent.

Case (b) (unboundedness of the linear form of the dual problem below and, consequently, inconsistency of restraints of the primal problem) arises if for some negative basis variable of the pseudoprogram $x_{r0} < 0$ all the coefficients x_{rj} are nonnegative.

Finally, case (c) is encountered if for any i, such that $x_{i0} < 0$, some of the x_{ij} are negative.

The first stage of the iteration thus terminates in one of the three possibilities (case (a), (b), or (c)). If case (a) or (b) holds, the solution process is terminated. In case (c), we proceed with the second stage of the iteration. The second stage entails determining the elementary transformation which will produce a new support program of the dual problem and, correspondingly, a new pseudoprogram of the primal problem with a lower value of the linear form. In the second stage we choose the position r of the dual basis with a negative pseudoprogram component and determine the vector A_k to be substituted for the vector A_{s_r} in the initial dual basis. The new dual basis is, thus, obtained from the old basis when the restraint vector A_{s_r} is replaced by A_k. In the second stage, moreover, we compute all the parameters required for determining and investigating a new pseudoprogram. The current pseudoprogram is analyzed in the first stage of the successive iteration.

§ 2. Geometrical interpretation

2-1. Following the two geometrical interpretations of the linear-programming problem we shall consider two geometrical interpretations of the dual method.

We start with the first geometrical interpretation. The polyhedral restraint set M of problem (1.1)–(1.3) is contained in the intersection of hyperplanes defined by equalities (1.2). The support programs of the problem correspond to vertices of the polyhedral set. Each vertex is formed by the intersection of n independent hyperplanes (m hyperplanes correspond to restraints (1.2) and $n-m$ hyperplanes correspond to zero components of the support program).

We give the geometrical interpretation of a pseudoprogram. The components of pseudoprogram X satisfy restraints (1.2) of the problem, all the extrabasis components x_j $(j \notin I_Y)$ being equal to zero. Moreover, the dual basis defining the pseudoprogram corresponds to nonnegative values of the parameters

$$\Delta_j = \sum_{i=1}^{m} c_{s_i} x_{ij} - c_j \geqslant 0$$

for $j = 1, 2, \ldots, n$. Hence, in the first geometrical interpretation, a pseudoprogram corresponds to a point X at which n independent hyperplanes intersect (m hyperplanes defined by restraints (1.2) and $n-m$ corresponding to the extrabasis components $x_j = 0$ of the pseudoprogram). In the general case, however, the point X corresponding to pseudoprogram X is located outside the polyhedral set M. To refine the geometrical definition of the pseudoprogram, we must establish the geometrical meaning of the conditions

$$\Delta_j \geqslant 0, \qquad j = 1, 2, \ldots, n.$$

We define a polyhedral cone K_X by

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, 2, \ldots, m, \tag{2.1}$$
$$x_j \geqslant 0, \quad j \notin I_Y = (s_1, s_2, \ldots, s_m).$$

The cone K_X is spanned by some of the hyperplanes of the polyhedral set M passing through the point X. Hence, the cone K_X has its apex at the point X.

We impose another condition on the geometrical analog of the pseudoprogram which amounts to requiring that the corresponding polyhedral cone K_X lie in the lower halfspace defined by the linear-form hyperplane passing through the point X. In other words, the vector C pointing in the direction of increasing linear form and the cone K_X should lie on either side of the linear-form hyperplane passing through the point X. We shall now show that this requirement corresponds to inequalities

$$\Delta_j \geqslant 0, \qquad j=1, 2, \ldots, n.$$

The last requirement together with

$$\sum_{\mu=1}^{m} a_{is_\mu} x_{s_\mu} = b_i, \qquad i=1, 2, \ldots, m.$$

$$x_j = 0, \qquad j \notin I_Y$$

defines the pseudoprogram.

Thus, let $S=\{A_{s_1}, \ldots, A_{s_m}\}$ be a linearly independent system of restraint vectors, and x_{s_l} and x_{lj} the coefficients in the expansion of the constraint vector B and the restraint vectors A_j in terms of the vectors of system S. Let, moreover, the s_i-th component of the n-dimensional vector X coincide with x_{s_l} $(l=1, \ldots, m)$, and all the other components of X be zero. Then, for any n-dimensional vector X',

$$L(X') = L(X) - \sum_{j \notin I_Y} \Delta_j x'_j, \tag{2.2}$$

where $I_Y=(s_1, \ldots, s_m)$. Relationship (2.2) is verified in the same way as formula (4.6) in Chapter 5, § 4.

Now, let the point X be a pseudoprogram of the problem, and let X' belong to cone K_X. Then

$$\Delta_j \geqslant 0, \qquad j=1, 2. \ldots, n$$

(X is a pseudoprogram), and

$$x'_j \geqslant 0, \quad j \notin I_Y \qquad (X' \in K_X).$$

We see from (2.2) that in this case

$$L(X') \leqslant L(X), \tag{2.3}$$

i. e., any point of the cone K_X and the vector C lies on either side of the linear-form hyperplane passing through the point X.

Conversely, let it be known that the cone K_X is in the lower halfspace, i. e., any point of the cone satisfies inequality (2.3). We take a point $X'=(x'_1, \ldots, x'_n)$ of the cone K_X such that

$$x'_\mu = \begin{cases} x_\mu - \theta x_{ij} & \text{for } \mu = s_i,\ i=1, 2, \ldots, m, \\ 0 & \text{for } \mu \notin I_Y,\ \mu \neq j, \\ \theta > 0 & \text{for } \mu = j. \end{cases} \tag{2.4}$$

Conditions (2.4) for $0 \leqslant \theta < \infty$ define the rays spanning cone K_X. It is easy to verify that for the point X'

$$L(X') = L(X) - \theta \Delta_j. \tag{2.5}$$

Comparing (2.3) and (2.5), we obtain

$$\Delta_j \geqslant 0.$$

We have thus established a correspondence between the geometrical image of the pseudoprogram and its analytical definition.

The point X, i.e., the geometrical analog of a pseudoprogram, will be called a *pseudovertex* of the polyhedral restraint set M.

It must be emphasized that a pseudovertex, like a pseudoprogram, is defined not only by the polyhedral restraint set, but also by the linear form of the problem. Moreover, we note that some of the hyperplanes constituting a pseudoprogram need not be boundary hyperplanes of the polyhedral set M. This refers to the hyperplanes corresponding to inequality restraints which follow from the other restraints.

2-2. Let us now geometrically interpret, in terms of the primal problem, the result of elementary transformation of the support program Y of the dual problem into $Y' = Y(\theta_0)$. In other words, we give a geometrical interpretation of the transformation of pseudoprogram X into pseudoprogram $X' = X(\theta_0)$.

We take the hyperplane $x_{s_r} = 0$ corresponding to the negative basis component x_{s_r} of pseudoprogram X.

Hyperplane $x_{s_r} = 0$ may have some points in common with the polyhedral cone K_X, though not necessarily. If the intersection of the cone and the hyperplane is an empty set, the problem is unsolvable, its restraints being inconsistent. Indeed, in this case the entire cone K_X is on the same side of hyperplane x_{s_r} as the pseudovertex X, for which $x_{s_r} < 0$. On the other hand, the cone K_X contains the polyhedral restraint set. Besides (2.1) the coordinates of the points in the domain of definition of the linear form of problem (1.1)–(1.3) also satisfy the condition $x_j \geqslant 0$ for $j \in I_Y$ and, in particular, $x_{s_r} \geqslant 0$. This contradiction will not occur only if the polyhedral restraint set is empty.

Now let the hyperplane $x_{s_r} = 0$ intersect the cone K_X. The intersection of the halfspace $x_{s_r} \leqslant 0$ and of the polyhedral cone K_X is the convex polyhedral set M_X. The pseudovertex X is one of its vertices. Each of the newly formed vertices is adjoining to the point X. Now let the linear-form hyperplane passing through the pseudovertex X undergo parallel translation until it touches the nearest vertex X' of the polyhedral set M_X. The first vertex of M_X reached by the linear-form hyperplane is a pseudovertex of the polyhedral set M, i.e., a geometrical image of the pseudoprogram corresponding to the support program Y'. The program $Y' = Y(\theta_0)$ is obtained from the support program Y of the dual problem under elementary transformation associated with the vector A_{s_r}.

We now prove the validity of the preceding geometrical interpretation of the transformation of pseudoprogram X into pseudoprogram X'.

Consider an arbitrary vertex of the polyhedral set M_X. This vertex is the intersection of m hyperplanes corresponding to equality restraints (1.2) of the problem with the hyperplane $x_{s_r} = 0$ and $n-m-1$ hyperplanes $x_j = 0\,(j \notin I_Y)$ corresponding to the extrabasis variables of pseudoprogram X. Since there is a total of $n-m$ extrabasis variables, M_X has at most $n-m$ vertices adjoining X. Let the vertex of M_X for which $x_t > 0\,(t \notin I_Y)$ be denoted by X_t. To obtain the components of X_t from the components of X, we must, following formulas (2.4) for $j = t$, increase θ until x'_{s_r} vanishes. To different vertices X_t

of the set M_X correspond the values θ given by

$$x'_{s_r}=x_{s_r}-\theta x_{rt}=0.$$

But $x_{s_r}<0$, $\theta>0$. This indicates that the vertex X_t exists only if $x_{rt}<0$. Hence, the values θ corresponding to the vertices X_t are computed from the condition

$$\theta=x_t=\frac{x_{s_r}}{x_{rt}} \quad \text{for} \quad x_{rt}<0.$$

The parameter $x_t\Delta_t$ coincides, up to a factor of normalization, with the distance of the point X_t from the linear-form hyperplane passing through the point X (see Exercise 2). Thus, closest to the linear-form hyperplane is the vertex X_k such that

$$x_k\Delta_k=\min_{x_{rt}<0} x_t\Delta_t=\min_{x_{rt}<0}\frac{x_{s_r}}{x_{rt}}\Delta_t.$$

The basis component x_{s_r} of pseudoprogram X is negative. Therefore,

$$x_k\Delta_k=|x_{s_r}|\min_{x_{rt}<0}\left(-\frac{\Delta_t}{x_{rt}}\right). \tag{2.6}$$

Therefore, the index k on which the ratio $-\frac{\Delta_t}{x_{rt}}$ with $x_{rt}<0$ attains its minimum defines the vertex $X=X_k$ of the polyhedral set M_X into which pseudoprogram X is transformed. Following these considerations we choose the index k of the vector to be introduced into the dual basis under elementary transformation of support program Y into $Y(\theta_0)$. The bases of the support programs Y and $Y'=Y(\theta_0)$ correspond to pseudoprograms X and X', respectively.

We have thus established that pseudoprogram X' corresponding to support program Y' is represented in this geometrical interpretation as the vertex X' of the polyhedral set M_X closest to X. The point X', like X, is a pseudovertex of the polyhedral restraint set M.

2-3. Let us now describe, in geometrical terms, a single iteration.

In the first stage of the iteration the pseudoprogram is tested for optimality. If pseudovertex X belongs to the polyhedral set M, the linear-form hyperplane passing through X proves to be the support hyperplane of the polyhedral set M at the point X, and the pseudoprogram X, being a feasible program of the problem, defines its solution (case (a)).

Now let program X not be a feasible program of the problem. Then, among its basis components we find $x_{s_r}<0$. For the hyperplane $x_{s_r}=0$ to intersect the cone K_X, it must intersect at least one of the rays (2.4) spanning the cone. In other words, for some t the conditions

$$x'_{s_r}=x_{s_r}-\theta x_{rt}=0$$

should be satisfied. But $x_{s_r}<0$, $\theta>0$. Therefore, $x'_{s_r}=0$ is impossible if all x_{rt} are negative. In this case, the hyperplane $x_{s_r}=0$ does not intersect the cone K_X. We have already seen that if the hyperplane $x_{s_r}=0$ and the cone K_X are disjoint, the linear-programming problem is unsolvable, its restraints being inconsistent (case (b)).

If the hyperplane $x_{s_r}=0$ intersects K_X, we proceed with the second stage of the iteration. The transformation from pseudoprogram X to pseudoprogram X' with a lower value of the linear form $L(X)$ corresponds to displacement from point X to the nearest vertex X' of the polyhedral set M_X

(case c). When passing from point X to X' the variable x'_{s_r} vanishes and the variable x_k (see (2.6)) is made a basis component of the pseudoprogram.

This parallel translation of the linear-form hyperplane decreases the value of the linear form in each successive iteration. Since the number of pseudoprograms is finite, the maximum of the linear form is obtained after a finite number of iterations. It is obvious from geometrical considerations that unsolvability of the problem (lack of feasible programs) will also be detected after a finite number of steps.

2-4. We shall now illustrate the preceding considerations using Example 1 from Chapter 4, § 3.

Example 1. Maximize the linear form

$$L(X)=5x_1-x_2-2x_3+5x_4+5x_5-x_6$$

subject to the conditions

$$\begin{aligned} -2x_1+5x_2+x_3 \qquad &=10, \\ x_1-x_2+x_4 \qquad &=1, \\ x_1+2x_2+x_5 \quad &=6, \\ 10x_1-3x_2+x_6 &=15, \\ x_j\geqslant 0,\ j=1,\ 2,\ \ldots,\ 6. \end{aligned}$$

Solution. The corresponding dual problem is stated as follows:
Minimize the linear form

$$\tilde{L}(Y)=10y_1+y_2+6y_3+15y_4$$

subject to the conditions

$$-2y_1+y_2+y_3+10y_4\geqslant 5, \tag{2.7}$$
$$5y_1-y_2+2y_3-3y_4\geqslant -1, \tag{2.8}$$
$$y_1 \geqslant -2, \tag{2.9}$$
$$y_2 \geqslant 5, \tag{2.10}$$
$$y_3 \geqslant 5, \tag{2.11}$$
$$y_4\geqslant -1. \tag{2.12}$$

We easily observe that replacing inequalities (2.7), (2.8), (2.11), (2.12) by equalities we obtain a system of four independent equations whose solution reduce restraints (2.9) and (2.10) to strict inequalities. Hence, the initial dual basis comprises the vectors A_1, A_2, A_5 and A_6*.

Let us now compute the basis components of the initial pseudoprogram and the coefficients in the expansion of the restraint vectors in terms of the vectors of the initial dual basis.

The basis components of the pseudoprogram satisfy the equations

$$B=\sum_{i=1}^{4} x_{i0}A_{s_i},$$

where $A_{s_1}=A_1$, $A_{s_2}=A_2$, $A_{s_3}=A_5$ and $A_{s_4}=A_6$. Thus, x_{i0} $(i=1,\ 2,\ 3,\ 4)$ are obtained from the system of equations

$$\begin{aligned} 10&=-2x_{10}+5x_{20}, \\ 1&=x_{10}-x_{20}, \\ 6&=x_{10}+2x_{20}+x_{30}, \\ 15&=10x_{10}-3x_{20}+x_{40}. \end{aligned}$$

Hence $x_{10}=5$, $x_{20}=4$, $x_{30}=-7$, $x_{40}=-23$ and, correspondingly,

$$X=(5,\ 4,\ 0,\ 0,\ -7,\ -23).$$

The linear form on pseudoprogram X is $L(X)=9$. Some of the pseudoprogram components are negative. Therefore, X is not a feasible program and as such does not define a solution of the problem.

Let us expand the restraint vectors in terms of the basis vectors of pseudoprogram X.

We have

$$A_j=\sum_{i=1}^{4} x_{ij}A_{s_i}.$$

* Methods for determining an initial support program of the dual problem are discussed in § 7.

Solving the corresponding systems of equations, we obtain the matrix

$$\|x_{ij}\| = \begin{Vmatrix} 1 & 0 & \frac{1}{3} & \frac{5}{3} & 0 & 0, \\ 0 & 1 & \frac{1}{3} & \frac{2}{3} & 0 & 0, \\ 0 & 0 & -1 & -3 & 1 & 0, \\ 0 & 0 & -\frac{7}{3} & -\frac{44}{3} & 0 & 1 \end{Vmatrix}$$

Some of the coefficients x_{3j} and x_{4j} corresponding to the negative basis components of the pseudoprogram are negative. There is, therefore, no reason to consider the problem unsolvable.

We have thus obtained case (c) from the dual basis and this enables us to pass to a successive pseudoprogram lowering the value of the linear form. The vector $A_{s_4} = A_6$ corresponding to the negative basis component of the pseudoprogram with highest absolute value is eliminated from the dual basis. The vector A_k to be introduced into the basis corresponds to the minimum θ_0 of the ratio $-\frac{\Delta_j}{x_{4j}}$ for negative x_{4j}. The parameters Δ_j are computed from (1.13):

$$\Delta_j = \sum_{i=1}^{m} c_{s_i} x_{ij} - c_j.$$

In our case

$$c_{s_1} = c_1 = 5, \quad c_{s_2} = c_2 = -1, \quad c_{s_3} = c_5 = 5, \quad c_{s_4} = c_6 = -1.$$

Therefore

$$\Delta_3 = 5 \cdot \frac{1}{3} + (-1)\frac{1}{3} + 5(-1) + (-1)\left(-\frac{7}{3}\right) - (-2) = \frac{2}{3},$$

$$\Delta_4 = 5 \cdot \frac{5}{3} + (-1)\frac{2}{3} + 5(-3) + (-1)\left(-\frac{44}{3}\right) - 5 = \frac{7}{3}.$$

We compute θ_0:

$$\theta_0 = \min\left(-\frac{\Delta_3}{x_{43}}, -\frac{\Delta_4}{x_{44}}\right) = \min\left(\frac{\frac{2}{3}}{\frac{7}{3}}, \frac{\frac{7}{3}}{\frac{44}{3}}\right) = \frac{7}{44}.$$

θ_0 is thus obtained on A_4. The vector A_4 should be introduced into the dual basis in place of $A_6 = A_{s_4}$. The vector A_4 is introduced into the fourth position and is accordingly denoted by A_{s_4}.

To compute the new pseudoprogram and the coefficients in the expansion of the restraint vectors in terms of the vectors of the new dual basis, we should solve systems of equations analogous to the preceding.

We have

$$X' = \left(\frac{105}{44}, \frac{65}{22}, 0, \frac{69}{44}, -\frac{101}{44}, 0\right).$$

$$\|x'_{ij}\| = \begin{Vmatrix} 1 & 0 & \frac{3}{44} & 0 & 0 & \frac{5}{44}, \\ 0 & 1 & \frac{5}{22} & 0 & 0 & \frac{1}{22}, \\ 0 & 0 & -\frac{23}{44} & 0 & 1 & -\frac{9}{44}, \\ 0 & 0 & \frac{7}{44} & 1 & 0 & -\frac{3}{44}. \end{Vmatrix}$$

The value of the linear form on pseudoprogram X' is $L(X') = \frac{235}{44}$. The signs of the component x_{30} of the pseudoprogram and of the coefficients x_{3j} indicate that we again have case (c). The vector $A_5 = A_{s_3}$ with a negative pseudoprogram component should be eliminated from the basis. It is replaced by the vector A_3 on which

$$\theta_0 = \min_{x_{3j}<0}\left(-\frac{\Delta_j}{x_{3j}}\right) = \min\left(\frac{\frac{13}{44}}{\frac{23}{44}}, \frac{\frac{7}{44}}{\frac{9}{44}}\right) = \frac{13}{23}$$

is obtained. All the coefficients x''_{i0} in the expansion of the constraint vector B in terms of the vectors

of the new basis are positive. We have thus obtained the optimal program

$$X'' = \left(\frac{48}{23},\ \frac{45}{23},\ \frac{101}{23},\ \frac{20}{23},\ 0,\ 0\right),$$
$$L(X'') = \frac{93}{23}.$$

2-5. We shall now use this example to give the geometrical interpretation of the solution of a linear-programming problem by the dual method.

As in Chapter 4, 4-2, we construct the problem equivalent to Example 1 which will enable us to interpret the principal stages of the iterations in the plane (x_1, x_2).

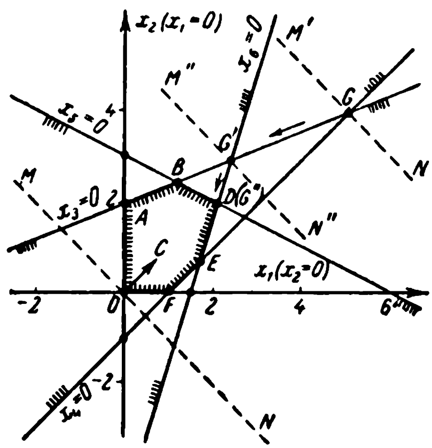


FIGURE 6.1

The equivalent problem is stated as follows:

Maximize the linear form

$$L(X) = x_1 + x_2$$

subject to the conditions

$$\begin{aligned} -2x_1 + 5x_2 &\leq 10, \\ x_1 - x_2 &\leq 1, \\ x_1 + 2x_2 &\leq 6, \\ 10x_1 - 3x_2 &\leq 15, \\ x_1 \geq 0,\quad x_2 &\geq 0. \end{aligned}$$

The equivalent problem (and consequently Example 1) corresponds to the domain of definition of the linear form shown in Figure 6.1. The sides of the polyhedron are defined by segments of the straight lines

$$\begin{aligned} x_1 = 0,\quad x_2 &= 0, \\ x_3 = 10 + 2x_1 - 5x_2 &= 0, \\ x_4 = 1 - x_1 + x_2 &= 0, \\ x_5 = 6 - x_1 - 2x_2 &= 0, \\ x_6 = 15 - 10x_1 + 3x_2 &= 0. \end{aligned}$$

Striation along the line indicates the halfplane in which the corresponding variable is nonnegative. The arrow OC defines the direction in which the linear-form line (hyperplane) is to be translated to increase $L(X)$.

The initial pseudoprogram $X = (5, 4, 0, 0, -7, -23)$ can easily be shown to correspond to the point $G(5,4)$ at which the lines $x_3 = 0$ and $x_4 = 0$ intersect. The point G is a pseudovertex of the polyhedral restraint vector M (polyhedron $OABDEF$). Indeed, the polyhedral cone K_X with its apex at the point X (in our case, the plane angle AGF) lies in the lower halfplane with respect to the linear-form line (hyperplane) $M'N'$ passing through the point G.

We will now illustrate, geometrically, the transformation to a current pseudoprogram. The line $x_6 = 0\ (x_{s_4} = 0)$ separates the pseudovertex G and the restraint polyhedron M. At point G the component $x_6 = x_{s_4} = -23 < 0$.

The line $x_6=0$ meets the cone K_X. The intersection of the halfplane $x_6 \leqslant 0$ and the cone K_X is a triangle $GO'E$ (polyhedral set M_X). The first vertex of the triangle reached when the line $M'N'$ is translated parallel to itself in the direction of decreasing linear form is G'. The point $G'=\left(\frac{105}{44}, \frac{65}{22}\right)$ is a pseudovertex of the polyhedron M, i. e., a geometrical image of the pseudoprogram

$$X'=\left(\frac{105}{44}, \frac{65}{22}, 0, \frac{69}{44}, -\frac{101}{44}, 0\right).$$

Analogous constructions enable us to pass from pseudovertex G' to point $D(G'')$ corresponding to pseudoprogram X''. Pseudovertex D is also a vertex of the restraint polyhedron. Pseudoprogram X'' is therefore a program, and consequently a solution, of the primal problem.

2-6. We now give an example terminating in case (b)—the linear-programming problem is unsolvable.

Example 2. Maximize the linear form

$$L(X)=5x_1+4x_2+x_3$$

subject to the conditions

$$\begin{aligned} x_1+2x_2-x_3 \qquad\quad &= 2, \\ 4x_1+3x_2 \qquad +x_4 \quad &= 12, \\ 4x_1+4x_2+x_3+x_4+x_5 &= 10, \\ x_j \geqslant 0, \quad j=1, \ldots, 5. & \end{aligned}$$

Solution. In the dual problem we minimize the linear form

$$\tilde{L}(Y)=2y_1+12y_2+10y_3$$

subject to the conditions

$$y_1+4y_2+4y_3 \geqslant 5, \tag{2.13}$$
$$2y_1+3y_2+4y_3 \geqslant 4, \tag{2.14}$$
$$-y_1 \qquad + y_3 \geqslant 0, \tag{2.15}$$
$$y_2+y_3 \geqslant 0, \tag{2.16}$$
$$y_3 \geqslant 1. \tag{2.17}$$

It can easily be seen that if in (2.13), (2.14), and (2.17) the inequalities are replaced by equalities, the solution of the system of three linearly independent equations will reduce restraints (2.15) and (2.16) to strict inequalities. Hence, the initial dual basis can be formed from the restraint vectors A_1, A_2, and A_5.

To obtain the corresponding pseudoprogram of the initial problem and the relevant parameters, we solve the system of equations

$$B=\sum_{i=1}^{3} x_{i0}A_{s_i},$$
$$A_3=\sum_{i=1}^{3} x_{i3}A_{s_i},$$
$$A_4=\sum_{i=1}^{3} x_{i4}A_{s_i}.$$

Here $A_{s_1}=A_1$, $A_{s_2}=A_2$, $A_{s_3}=A_5$. We have

$$X=\left(\frac{18}{5}, -\frac{4}{5}, 0, 0, -\frac{6}{5}\right),$$

$$\|x_{ij}\|=\left\|\begin{matrix} 1 & 0 & \frac{3}{5} & \frac{8}{5} & 0, \\ 0 & 1 & -\frac{4}{5} & -\frac{1}{5} & 0, \\ 0 & 0 & \frac{9}{5} & \frac{1}{5} & 1 \end{matrix}\right\|.$$

The basis components x_{20} and x_{30} of the pseudoprogram X are negative, and the expansion coefficients x_{3j} of all the restraint vectors in the dual basis, associated with the vector A_5, are nonnegative.

We thus have case (b). The problem is unsolvable, its restraints being inconsistent.

To obtain a geometrical interpretation of case (b), we replace, as in Example 1, the given problem by the equivalent problem from which the negative variables x_3, x_4, x_5 have been eliminated and the equality restraints replaced by inequalities.

The equivalent problem is stated as follows:

Maximize the linear form

$$L(X) = 4x_1 + x_2$$

subject to the conditions

$$\begin{gathered} x_1 + 2x_2 \geqslant 2, \\ 4x_1 + 3x_2 \leqslant 12, \\ x_1 + 3x_2 \leqslant 0, \\ x_1 \geqslant 0, \quad x_2 \geqslant 0. \end{gathered}$$

The sides of the restraint polyhedron M defining the domain of definition of the linear form are defined by the lines (Figure 6.2)

$$\begin{aligned} x_1 &= 0, \quad x_2 = 0, \\ x_3 &= -2 + x_1 + 2x_2 = 0, \\ x_4 &= 12 - 4x_1 - 3x_2 = 0, \\ x_5 &= \quad - x_1 - 3x_2 = 0. \end{aligned}$$

The striation along the lines and the arrow OC serve the same purpose as in the preceding example.

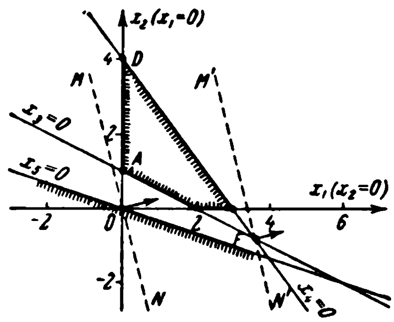


FIGURE 6.2

It is easy to verify directly that the initial pseudoprogram $X = \left(\frac{18}{5}, -\frac{4}{5}, 0, 0, -\frac{6}{5}\right)$ corresponds to the point of intersection $F\left(\frac{18}{5}, -\frac{4}{5}\right)$ of the lines $x_3 = 0$ and $x_4 = 0$. The cone K_X in our case coincides with the plane angle AFD. The cone K_X lies in the lower halfplane with respect to the linear-form line $M'N'$ drawn through the point F. The point F is, thus, a pseudovertex.

The line $x_5 = 0$ corresponding to the negative component of the pseudoprogram $\left(x_5 = x_{i0} = -\frac{6}{5}\right)$ does not intersect the cone K_X (the plane angle AFD). This is possible, as we have seen, only if the restraint polyhedron (the domain of definition of the linear form) is an empty set.

The above argument is given in order to illustrate case (b) in the dual simplex procedure and give its geometrical interpretation. In fact, however, unsolvability of Example 2, i.e., inconsistency of the problem restraints, is immediately obvious from the restraints of the equivalent problem, and also from Figure 6.2.

2-7. We now give a geometrical interpretation of the dual method corresponding to the second geometrical interpretation of the linear-programming problem.

In Chapter 3, 1-2, we described the geometry of a pair of dual problems in the $(m+1)$-dimensional space of points $U = (u_1, \ldots, u_{m+1})$. We remind the reader that the augmented restraint vectors $\overline{A}_j$ $(j = 1, 2, \ldots, n)$ span, in the $(m+1)$-dimensional U-space, a convex polyhedral cone K corresponding to the set of n-dimensional vectors with nonnegative components. Moreover, in the U-space, we consider the line Q such that

$$\begin{aligned} u_i &= b_i, & i &= 1, 2, \ldots, m, \\ u_{m+1} &= t, & -\infty &< t < \infty. \end{aligned}$$

To feasible programs X of the primal problem correspond those and only those points U which belong to the cone K and to the line Q. To the

optimal program of the problem corresponds the highest (in the sense of the Ou_{m+1}-axis) point of intersection of the line Q and the cone K.

The geometrical image of the set of feasible programs of the dual problem in the $(m+1)$-dimensional U-space coincides with the set of all hyperplanes passing through the origin and extending above the cone K. (The cone K and the vector $(0, \ldots, 0, 1)$ lie on opposite sides of the hyperplanes.) The optimal program of the dual problem corresponds to the hyperplane passing through the origin and the highest point of intersection of the line Q and the cone K and extending above the cone K.

Let us now clarify the geometrical meaning of the dual method. The augmented basis vectors of the initial support program of the dual problem define a hyperplane Π passing through the origin and extending above the cone K. The hyperplane Π contains m augmented vectors $\overline{A}_j$ and, hence, contains some m-dimensional face of the cone K. If the vector corresponding to the highest point of intersection of the line Q and the cone K belongs to this face, the hyperplane Π specifies the optimal program of the dual problem (case (a)).

Now let the line Q meet the hyperplane Π at point X above the highest point of intersection of the line Q and the cone K. The point X corresponds to some pseudoprogram of the primal problem. Let some basis component x_{s_r} of the pseudoprogram be negative, and all the expansion coefficients x_{rj} of the restraint vectors in the dual basis associated with the vector A_{s_r} be nonnegative. This indicates that in the halfspace $u_{m+1}=0$ we may construct a $(m-1)$-dimensional hyperplane H_{m-1} separating the point B (specified by the constraint vector) and all the points A_j (specified by the restraint vectors). The hyperplane H_{m-1} of the m-dimensional space $u_{m+1}=0$ is spanned by the basis vectors $A_{s_1}, A_{s_2}, \ldots, A_{s_{r-1}}, A_{s_{r+1}}, \ldots, A_{s_m}$. The equation of the hyperplane H_{m-1} can be written in the form $(A, \tilde{U})=0$, where $\tilde{U}$ is a point in the halfspace $u_{m+1}=0$, and the vector A is orthogonal to the vectors $A_{s_1}, \ldots, A_{s_{r-1}}, A_{s_{r+1}}, \ldots, A_{s_m}$, i. e., $(A, A_{s_i})=0$ for $i=1, \ldots, r-1, r+1, \ldots, m$. To be specific, let $(A, A_{s_r})=1$. Computing the scalar products (A, A_j) and (A, B) we have

$$(A, A_j)=\sum_{i=1}^{m} x_{ij}(A, A_{s_i})=x_{rj}(A, A_{s_r})=x_{rj},$$

$$(A, B)=\sum_{i=1}^{m} x_{i0}(A, A_{s_i})=x_{r0}.$$

But $x_{r0}<0$, $x_{rj}>0$. Hence, the points A_j and the point B are on opposite sides of the hyperplane $(A, \tilde{U})=0$.

In the $(m+1)$-dimensional U-space consider the hyperplane H_m spanned by the vectors $A_{s_1}, \ldots, A_{s_{r-1}}, A_{s_{r+1}}, \ldots, A_{s_m}$ and the vector e_{m+1}. The hyperplane H_m separates the line Q and the cone K spanned by the augmented vectors A_j. This follows from the fact that H_{m-1} separates the constraint vector and the restraint vectors of the problem in the subspace $u_{m+1}=0$. Hence, in our case, the line Q and the cone K are disjoint. The primal problem has no feasible programs (case (b)).

Now let the line Q meet the cone. Let $\overline{A}_{s_r}$ be the augmented restraint vector corresponding to a negative component of the pseudoprogram. We now rotate the hyperplane Π about the vectors $\overline{A}_{s_1}, \ldots, \overline{A}_{s_{r-1}}, \overline{A}_{s_{r+1}}, \ldots, \overline{A}_{s_m}$ so that the point of intersection of the hyperplane and the line Q descends (in the sense of the Ou_{m+1}-axis). The rotation is continued until the hyperplane Π captures one of the augmented vectors A_j, e. g., A_k. The vectors $\overline{A}_{s_1}, \ldots$

$\ldots, \bar{A}_{s_{r-1}}, \bar{A}_k, \bar{A}_{s_{r+1}}, \ldots, \bar{A}_{s_m}$ constitute the next dual basis. The hyperplane Π_1 defined by these vectors corresponds to the new support program of the dual problem. The point of intersection of hyperplane Π_1 and the line Q is the image of the current pseudoprogram X'. The value of the linear form $L(X)$ on pseudoprogram X' is less than on pseudoprogram X (the point X' is below the point X). If after testing the pseudoprogram X' for optimality and the problem for unsolvability we do not obtain cases (a) or (b), we apply to hyperplane Π_1 a procedure analogous to that previously applied to hyperplane Π. Successive rotations of hyperplanes, corresponding to successive support programs of the dual problem, eventually produce a hyperplane Π^* meeting the line Q at the point X^*, which is a boundary point of the cone K. The hyperplane corresponds to the optimal support program of the dual problem, and the pseudoprogram X^* is a program and, consequently, a solution of the primal problem.

In the dual simplex method we thus approach the optimal program of the primal problem (the highest point of intersection of the line Q and the cone K) not from within the cone, as in the simplex method, but from without.

2-8. We shall now illustrate geometrically the application of the dual method to the determination of an optimal program of the problem solved in Chapter 4, 4-3, by the simplex method.

The problem is:

Maximize the linear form

$$L(X) = 10x_1 + 8x_2 + 7x_3 + 16x_4 + 21x_5$$

subject to the conditions

$$\begin{gathered} 4x_1 + 2x_2 + 5x_3 + 10x_4 + 5x_5 = 6, \\ 9x_1 + 10x_2 + 12.5x_3 + 18x_4 + 16.5x_5 = 14, \\ x_j \geqslant 0, \quad j = 1, 2, \ldots, 5. \end{gathered}$$

In Figure 4.3, Chapter 4, the vectors A_j and the constraint vector B, the corresponding augmented vectors, and the line Q passing through the point B parallel to the Ou_3-axis are shown.

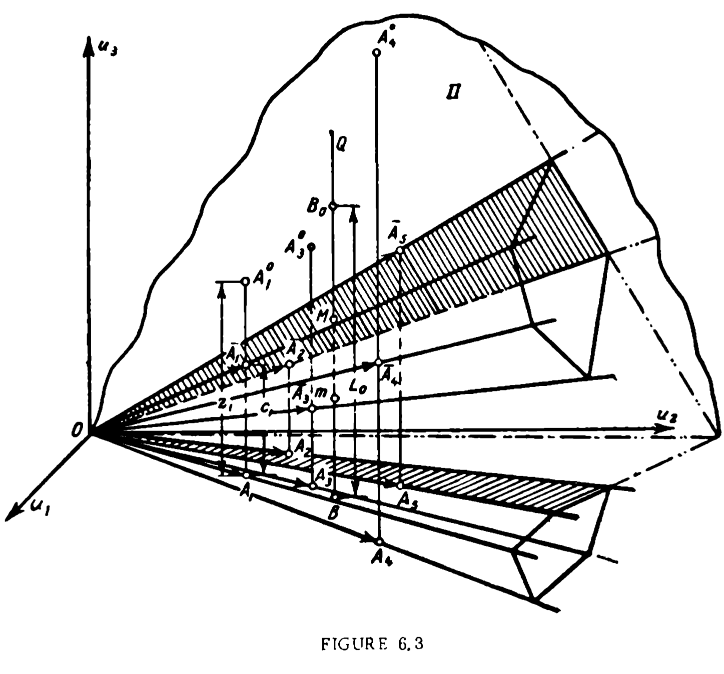


FIGURE 6.3

The plane Π defined by the augmented vectors $\bar{A}_2$ and $\bar{A}_5$ passes through the origin and extends above the cone K spanned by the augmented restraint vectors. Hence, the plane Π is the image of a support program of the dual problem, and the point of intersection B_0 of the plane Π and the line Q corresponds to pseudoprogram X with the basis (A_2, A_5).

In Figure 6.3 the plane Π spanned by the augmented restraint vectors $\bar{A}_2$ and $\bar{A}_5$ is shown. The two-dimensional cones (plane angles) spanned by the augmented vectors $\bar{A}_2'$ and $\bar{A}_5'$ and the restraint vectors A_2 and A_5 are striated.

The point B_0 is outside the striated cone (angle), i. e. , the component x_2 of pseudoprogram X is negative. The vector A_2 should be eliminated from the basis. We rotate the plane Π about the vector $\bar{A}_5$ so that the point of intersection of the plane and the line Q descends. The first vector to be captured in this rotation is the vector $\bar{A}_1$. The plane Π_1 formed by the vectors $\bar{A}_1$ and $\bar{A}_5$ is the image of the successive support program of the problem (Figure 6.4).

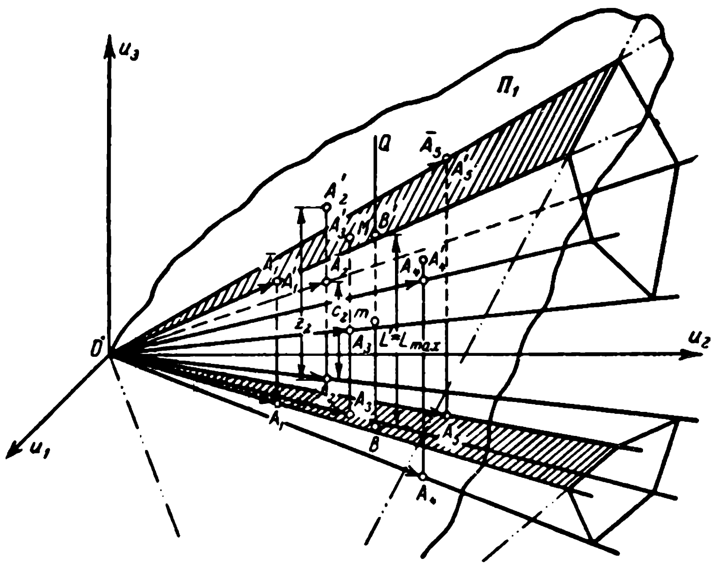


FIGURE 6.4

The plane Π_1 meets the line Q at the point B' which is a boundary point of the cone K. The point B' is, thus, the image of the optimal program of the problem. The basis components of the support solution coincide with the coefficients in the expansion of the constraint vector B in terms of the restraint vectors A_1 and A_5. We have $x'_{10}=\frac{29}{21}$ and $x'_{50}=\frac{2}{21}$. The optimal program of the problem is, thus,

$$X'=\left(\frac{29}{21},\, 0,\, 0,\, 0,\, \frac{2}{21}\right).$$

The maximum possible value of the linear form is

$$L(X')=B'B=\frac{332}{21}.$$

The components y_1, y_2 of the optimal program Y of the dual problem coincide with the first $m=2$ components of the direction vector of the plane Π_1 (the vector $\bar{Y}=(y_1, y_2, \ldots, y_m, y_{m+1})$ orthogonal to Π_1 and normalized by the requirement $y_{m+1}=-1$).

The equation of the plane Π_1 is

$$y_1u_1+y_2u_2=u_3.$$

The points $\bar{A}_1(a_{11}, a_{21}, c_1)$ and $\bar{A}_5(a_{15}, a_{25}, c_5)$ belong to the plane Π_1. Therefore

$$y_1a_{11}+y_2a_{21}=c_1, \quad y_1a_{15}+y_2a_{25}=c_5.$$

From these equations we compute the components y_1 and y_2 of the solution of the dual problem. In our case

$$\bar{A}_1=(4;\ 9;\ 10),\quad \bar{A}_2=(5;\ 16.5;\ 21),$$

so that the optimal program of the dual problem coincides with the vector $Y=\left(-\frac{24}{21},\ \frac{34}{21}\right)$.

§ 3. The case of bilateral restraints

3-1. Consider a linear-programming problem with bilateral restraints: Maximize the linear form

$$L(X)=\sum_{j=1}^{n} c_j x_j \tag{3.1}$$

subject to the conditions

$$\sum_{j=1}^{m} A_j x_j=B, \tag{3.2}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j. \tag{3.3}$$

Here $A_j=(a_{1j},\ a_{2j},\ \ldots,\ a_{mj})^T$ and $B=(b_1,\ b_2,\ \ldots,\ b_m)^T$. As always, the rank of matrix $(A_1,\ A_2,\ \ldots,\ A_n)$ is taken as equal to m.

Problem (3.1)-(3.3) was discussed in Chapter 4, § 5, where it was solved by the simplex method. Now, as then, we shall not assume all the variables to be bilaterally restrained. Some of them may have only one boundary value. In this case, the other boundary value (α_j or β_j) is assumed infinite ($\alpha_j=-\infty$ or $\beta_j=\infty$).

In this section we describe the dual simplex method as applied to problems with bilateral restraints.

3-2. Let us state the dual problem with respect to problem (3.1)-(3.3). According to the general principles of duality (see Chapter 3, 4-1), this problem entails minimization of the linear form

$$\tilde{L}(Y,\ Z',\ Z'')=\sum_{i=1}^{m} b_i y_i-\sum_{j\in I'} \alpha_j z'_j+\sum_{j\in I''} \beta_j z''_j \tag{3.4}$$

subject to the conditions

$$\sum_{i=1}^{m} a_{ij} y_i-\delta'_j z'_j+\delta''_j z''_j=c_j,\qquad j=1,\ 2,\ \ldots,\ n; \tag{3.5}$$

$$z'_j\geqslant 0,\quad j\in I';\quad z''_j\geqslant 0,\quad j\in I''. \tag{3.6}$$

Here I' (I'') is the set of indices j for which $\alpha_j>-\infty$ ($\beta_j<\infty$),

$$\delta'_j=\begin{cases}1, & \text{if } j\in I',\\ 0, & \text{if } j\notin I';\end{cases}$$

$$\delta''_j=\begin{cases}1, & \text{if } j\in I'',\\ 0, & \text{if } j\notin I''.\end{cases}$$

It is left to the reader to verify that problem (3.4)-(3.6) is indeed dual with respect to problem (3.1)-(3.3) (see Exercise 3).

Let the vector $(Y;\ Z';\ Z'')$, where

$$Y=(y_1,\ y_2,\ \ldots,\ y_m),\quad Z'=\{z'_i\}_{i\in I'},\quad Z''=\{z''_i\}_{i\in I''},$$

be a feasible program of the dual problem. Let

$$\Delta_j=\sum_{i=1}^{m} a_{ij} y_i-c_j,\qquad j=1,\ 2,\ \ldots,\ n. \tag{3.7'}$$

Then

$$(A_j, Y) - c_j = \begin{cases} \Delta_j \geqslant 0, & \text{if } j \notin I'' \ (\beta_j = \infty, \ \alpha_j > -\infty), \\ \Delta_j \leqslant 0, & \text{if } j \notin I' \ (\alpha_j = -\infty, \ \beta_j < \infty). \end{cases} \tag{3.7}$$

Indeed, for $j \notin I''$, we have from (3.5)

$$\Delta_j = \delta'_j z'_j - \delta''_j z''_j = \delta'_j z'_j \geqslant 0.$$

The second requirement of system (3.7) is verified in the same way.

Now let $Y = (y_1, y_2, \ldots, y_m)$ be an m-dimensional vector satisfying conditions (3.7). Then, there exist z'_j, $j \in I'$, and z''_j, $j \in I''$, such that the vector (Y, Z', Z'') is a feasible program of problem (3.4)–(3.6). Indeed, it suffices to take

$$z'_j = \begin{cases} \Delta_j, & \text{if } \Delta_j \geqslant 0, \\ 0, & \text{if } \Delta_j < 0 \end{cases} \quad (j \in I'), \tag{3.8}$$

$$z''_j = \begin{cases} -\Delta_j, & \text{if } \Delta_j \leqslant 0, \\ 0, & \text{if } \Delta_j > 0 \end{cases} \quad (j \in I''). \tag{3.9}$$

If $j \notin I'$, $j \in I''$, then, from (3.7), $\Delta_j \leqslant 0$, and correspondingly

$$(A_j, Y) - \delta'_j z'_j + \delta''_j z''_j = (A_j, Y) + z''_j = (A_j, Y) - \Delta_j = c_j.$$

The case $j \in I'$, $j \notin I''$ is verified in the same way. For $j \in I'$, $j \in I''$ (the variable x_j of the primal problem has two finite boundary values)

$$(A_j, Y) - \delta'_j z'_j + \delta''_j z''_j = (A_j, Y) - z'_j + z''_j.$$

From the definition of z'_j and z''_j ($j \in I'$, $j \in I''$)

$$z'_j - z''_j = \Delta_j.$$

Hence, the vector (Y, Z', Z'') satisfies the j-th equality in (3.5). Observe that the case $j \notin I'$, $j \notin I''$ is impossible, because each variable of the primal problem is assumed bounded at least on one side. By definition, $z'_j \geqslant 0$, $z''_j \geqslant 0$. The vector (Y, Z', Z'') is indeed a feasible program of the dual problem.

With fixed Y satisfying (3.7), z'_j for $j \notin I''$ and z''_j for $j \notin I'$ are uniquely defined:

$$z'_j = \Delta_j; \quad z''_j = -\Delta_j.$$

Now if $j \in I'$, $j \in I''$, the j-th equality in (3.5) is satisfied, as we have already seen, if

$$z'_j - z''_j = \Delta_j. \tag{3.10}$$

Hence, if there exist bilaterally restrained variables x_j, to each vector Y satisfying conditions (3.7) there correspond several feasible programs of the dual problem.

From these programs we choose that which is associated with the least value of the linear form (3.4). For any $j \in I'$, I'', we find negative z'_j and z''_j satisfying equality (3.10) and such that $-\alpha_j z'_j + \beta_j z''_j$ is conditionally minimized. It is easily observed that these z'_j and z''_j are determined from (3.8) and (3.9):

$$z'_j = \begin{cases} \Delta_j, & \text{if } \Delta_j \geqslant 0, \\ 0, & \text{if } \Delta_j < 0; \end{cases}$$

$$z''_j = \begin{cases} -\Delta_j, & \text{if } \Delta_j \leqslant 0, \\ 0, & \text{if } \Delta_j > 0. \end{cases}$$

Thus, the z'_j, $j \in I'$, z''_j, $j \in I''$, which, together with the fixed vector Y, specify a program of problem (3.4)–(3.6) minimizing $\tilde{L}(Y, Z', Z'')$ are computed from (3.8) and (3.9).

Inserting these z'_j and z''_j into the linear form of the dual problem, we

have

$$\tilde{L}(Y)=\tilde{L}(Y,\ Z',\ Z'')=\sum_{i=1}^{m} b_i y_i-\sum_{j=1}^{n}\gamma_j\Delta_j, \tag{3.11}$$

where

$$\gamma_j=\begin{cases}\alpha_j, & \text{if } \Delta_j\geqslant 0,\\ \beta_j, & \text{if } \Delta_j<0.\end{cases}$$

Since we are concerned with programs of the dual problem corresponding to the least possible value of the linear form (3.4), we shall consider only those of the programs whose components z'_j and z''_j are uniquely determined by (3.8), (3.9). Remembering this, we may consider instead of the dual-problem programs the vectors

$$Y=(y_1,\ y_2,\ \ldots,\ y_m),$$

satisfying conditions (3.7).

3-3. We introduoe some definitions.

A vector $Y=(y_1,\ y_2,\ \ldots,\ y_m)$ satisfying conditions (3.7) is called the *program kernel* of problem (3.4)–(3.6).

Applying the definition of a support program given in Chapter 2, § 5, we may verify that the vector Y is a kernel of a *support* program of problem (3.4)–(3.6) if and only if conditions (3.7) are satisfied and among the vectors A_j, $j=1, 2, \ldots, n$, there exist m linearly independent vectors A_{s_1}, A_{s_2}, ..., A_{s_m} such that

$$\Delta_{s_i}=(A_{s_i},\ Y)-c_{s_i}=0, \qquad i=1,\ 2,\ \ldots,\ m \tag{3.12}$$

(see Exercise 4).

The system of linearly independent restraint vectors A_{s_i}, $i=1, 2, \ldots, m$ for which (3.12) applies is called the *basis* of kernel Y of a support program of the dual problem.

The set of indices of the vectors A_j constituting the basis of kernel Y will be denoted by I_Y. In our case $I_Y=(s_1, s_2, \ldots, s_m)$.

It is easy to show that the support program $(Y,\ Z',\ Z'')$ of problem (3.4)–(3.6) is nondegenerate if and only if its kernel satisfies the condition

$$\Delta_j=(A_j,\ Y)-c_j\neq 0 \ \text{ for } \ j\notin I_Y \tag{3.13}$$

(see Exercise 5).

In this section the dual problem (3.4)–(3.6) is assumed to be nondegenerate. The kernels of all the support programs of the problem should, therefore, satisfy condition (3.13).

As in the case of the linear-programming problem in canonical form, it is advisable to make each support program of the dual problem correspond to some n-dimensional vector X satisfying equations (3.2).

Let $Y=(y_1,\ y_2,\ \ldots, y_m)$ be the kernel of a support program of problem (3.4)–(3.6). We define the vector $X=(x_1,\ x_2,\ \ldots,\ x_n)$ as follows:

(a) if $j\notin I_Y$,

$$x_j=\begin{cases}\alpha_j \text{ for } \Delta_j>0,\\ \beta_j \text{ for } \Delta_j<0;\end{cases} \tag{3.14}$$

(b) the other components of the vector X are computed from

$$\sum_{j\in I_Y} x_j A_j=B-\sum_{j\notin I_Y} x_j A_j. \tag{3.15}$$

The vector X obviously satisfies restraints (3.2) of the primal problem.

The vector X associated with the kernel Y of a support program of the dual problem (or, more precisely, with the basis of this kernel) by relationships (3.14) and (3.15) will be called a p s e u d o p r o g r a m of the primal problem (3.1)–(3.3).

Let X be the pseudoprogram corresponding to kernel Y. The system of vectors A_j, $j \in I_Y$, constituting the basis of kernel Y is sometimes called the b a s i s o f p s e u d o p r o g r a m X.

The components x_j of pseudoprogram X are called b a s i s components for $j \in I_Y$, the remaining being e x t r a b a s i s variables.

In the nondegenerate case, to each kernel of a support program there corresponds a unique basis and, consequently, a unique pseudoprogram. In the degenerate case come kernels may have several bases and, consequently, correspond to several pseudoprograms. The application of the dual method to a problem with bilateral restraints involves passing from one support-program kernel of the dual problem to another (or, equivalently, from one pseudoprogram of the primal problem to another) monotonically decreasing linear form (3.4).

3-4. The principal feature of the dual method, as of any other linear-programming method, is the test for finding the solution of the problem.

Optimality test. *If pseudoprogram X is a feasible program of the primal problem, i.e., if its basis components satisfy restraints (3.3), it solves this problem. The corresponding vector Y is then the kernel of the optimal program of the dual problem.*

Proof of the first part of this proposition follows directly from the second form of the optimality test established in Chapter 4, § 5, since according to (3.14) and (3.15) program X fulfills all the requirements of this test. We shall require only the first part of the optimality test. As regards the second part of the test referring to the kernel Y, its proof is left to the reader (see Exercise 6).

Consider the kernel $Y=(y_1, y_2, \dots, y_m)$ of some support program of the dual problem.

Let the restraint vectors $A_{s_1}, A_{s_2}, \dots, A_{s_m}$ constitute the basis of this kernel, so that $I_Y=(s_1, s_2, \dots, s_m)$. Applying formulas (3.14) and (3.15) we associate pseudoprogram $X=(x_1, x_2, \dots, x_n)$ with the kernel Y and its basis A_j, $j \in I_Y$. The system of vectors $A_j, j \in I_Y$, constitutes the basis of pseudoprogram X.

We now expand in terms the basis vectors of pseudoprogram X all the restraint vectors A_j and the vector

$$A_0 = B - \sum_{j \in I_Y} x_j A_j.$$

We have

$$A_j = \sum_{i=1}^{n} x_{ij} A_{s_i}, \qquad j=0, 1, \dots, n. \tag{3.16}$$

As always, x_{ij} denotes the expansion coefficient of the vector A_j associated with the basis vector A_{s_i} occupying the i-th position in the given basis. In particular, for $j=0$, the expansion coefficients in (3.16) coincide with the basis components of the pseudoprogram, i.e., $x_{s_i} = x_{i0}$, $i=1, 2, \dots, m$. According to (3.7')

$$\Delta_j = \sum_{i=1}^{m} a_{ij} y_i - c_j,$$

where $Y=(y_1, y_2, \ldots, y_m)$ is the support-program kernel of the dual problem. By assumption, $\Delta_j=0$ for $j\in I_Y$.

In § 1 we showed (see (1.13)) that the parameters Δ_j can also be computed from the x_{ij}:

$$\Delta_j=\sum_{i=1}^{m} c_{s_i} x_{ij}-c_j. \tag{3.17}$$

Given the parameters x_{ij} and Δ_j computed from the given kernel Y (or, equivalently, from the corresponding pseudoprogram X), we may establish optimality of X (if it is indeed optimal) or construct a new pseudoprogram X' corresponding to a kernel Y' which is associated with a lower value of linear form (3.4) than Y. These parameters also enable us to establish unsolvability of the primal problem. As in the canonical case, there too three mutually exclusive cases depending on the relative signs of the parameters x_{ij} and Δ_j arise:

(a) All the basis components of pseudoprogram X satisfy restraint (3.3), i. e.,

$$\alpha_{s_i}\leqslant x_{s_i}=x_{i0}\leqslant\beta_{s_i} \quad \text{for} \quad i=1, 2, \ldots, m. \tag{3.18}$$

(b) Some of the basis components do not satisfy conditions (3.18). Among these there is a component $x_{i_0}=x_{s_i}$ such that

$$x_{i0}<\alpha_{s_i},$$
$$x_{ij}\begin{cases} \geqslant 0 \text{ for } \Delta_j>0, \\ \leqslant 0 \text{ for } \Delta_j<0 \end{cases} \quad (j\notin I_Y) \tag{3.19}$$

or

$$x_{i0}>\beta_{s_i},$$
$$x_{ij}\begin{cases} \leqslant 0 \text{ for } \Delta_j>0, \\ \geqslant 0 \text{ for } \Delta_j<0 \end{cases} \quad (j\notin I_Y). \tag{3.20}$$

(c) Among the basis components of pseudoprogram X there are some which do not satisfy restraints (3.18). However, for $x_{i0}<\alpha_{s_i}$, (3.19) do not hold, and for $x_{i0}>\beta_{s_i}$, (3.20) do not apply.

If conditions (3.18) are satisfied (case (a)), pseudoprogram X meets the requirements of the optimality test. Hence, the vector X solves the primal problem.

To analyze the two other cases, we find it expedient to use a concept analogous to elementary transformation of a dual-problem program discussed in § 1.

3-5. As in § 1, $\|e_{ij}\|_m$ denotes the matrix constructed of the coefficients in the expansion of the unit vectors $e_1, e_2, \ldots, e_m$ in terms of the basis vectors of kernel Y (the basis of pseudoprogram X). Thus,

$$\|e_{ij}\|_m=(A_{s_1}, A_{s_2}, \ldots, A_{s_m})^{-1}.$$

To each restraint vector A_{s_l}, $1\leqslant l\leqslant m$, we associate a transformation of the support-program kernel Y of the dual problem defined by

$$Y(\theta)=(y_1(\theta), y_2(\theta), \ldots, y_m(\theta))=Y+\theta e^{(l)}, \tag{3.21}$$

where $e^{(l)}=(e_{l1}, e_{l2}, \ldots, e_{lm})$ is the m-dimensional vector comprising the elements of the l-th row of matrix $\|e_{ij}\|_m$. This transformation is *the elementary transformation associated with the vector* A_{s_l}.

We now investigate the relevant properties of the vector $Y(\theta)$ for different θ. To this end we consider

$$\Delta_j(\theta)=\sum_{\lambda=1}^{m} a_{\lambda j} y_\lambda(\theta)-c_j.$$

From (3.21)

$$\Delta_j(\theta)=\sum_{\lambda=1}^{m} a_{\lambda j} y_\lambda+\theta\sum_{\lambda=1}^{m} a_{\lambda j} e_{i\lambda}-c_j=\Delta_j+\theta\sum_{\lambda=1}^{m} a_{\lambda j} e_{i\lambda}. \tag{3.22}$$

Since the vector $X_j=(x_{1j}, x_{2j}, \ldots, x_{mj})^T$ is a solution of the system of linear equations

$$(A_{s_1}, A_{s_2}, \ldots, A_{s_m}) X_j=A_j$$

(see (3.16)), we obtain

$$X_j=(A_{s_1}, A_{s_2}, \ldots, A_{s_m})^{-1}A_j=\| e_{i\lambda} \|_m A_j.$$

Hence

$$x_{ij}=\sum_{\lambda=1}^{m} a_{\lambda j} e_{i\lambda}, \tag{3.23}$$

$$i=1, 2, \ldots, m; \quad j=0, 1, 2, \ldots, n.$$

Applying (3.22) and (3.23), we have

$$\Delta_j(\theta)=\Delta_j+\theta x_{ij}. \tag{3.24}$$

In the following, we shall have to distinguish between two cases

$$\text{i) } \theta\geqslant 0, \quad \text{ii) } \theta\leqslant 0.$$

Consider (i). Let θ increase from 0 on upwards as long as all $\Delta_j(\theta)$ maintain their respective signs. In other words, θ should satisfy the conditions

$$\Delta_j(\theta)=\Delta_j+\theta x_{ij}\begin{cases} \geqslant 0, & \text{if } \Delta_j>0, \\ \leqslant 0, & \text{if } \Delta_j<0. \end{cases}$$

(From the assumption of nondegeneracy $\Delta_j\neq 0$ for $j\notin I_Y$.) A departure from the first of these conditions is possible only if $x_{ij}<0$, and from the second only if $x_{ij}>0$. The two conditions are, therefore, equivalent to the inequality

$$\theta\leqslant\theta_0=\min_j\left(-\frac{\Delta_j}{x_{ij}}\right), \tag{3.25}$$

where the minimum is taken over all j for which $\frac{\Delta_j}{x_{ij}}<0$, $x_{ij}\neq 0$. If for any $j\notin I_Y$, Δ_j and x_{ij} $(x_{ij}\neq 0)$ have the same sign, $\theta_0=\infty$. Observe that since the dual-problem program with the kernel Y is assumed nondegenerate, $\theta_0>0$.

Thus, in the first case

$$0\leqslant\theta\leqslant\theta_0,$$

where θ_0 is defined by (3.25) and $\theta_0=\infty$ if the set of indices j on which the minimum in (3.25) is taken as empty.

Consider (ii). Here θ should be nonpositive. We again determine the limit of θ for which the signs of $\Delta_j(\theta)$ and Δ_j coincide for all $j\notin I_Y$. In the same way as above we obtain

$$-\theta\leqslant\theta_0=\min_j\frac{\Delta_j}{x_{ij}}, \tag{3.26}$$

where the minimum is taken over those $j\notin I_Y$ for which $\frac{\Delta_j}{x_{ij}}>0$, $x_{ij}\neq 0$. When none of the indices j have this property, θ_0 is taken equal to ∞.

Thus, for (ii)

$$0\leqslant-\theta\leqslant\theta_0,$$

where θ_0 is determined following the preceding rules. In the nondegenerate case $\theta_0>0$.

The elementary transformation associated with the vector A_{s_i} is applied with $\theta\geqslant 0$ only if $\alpha_{s_i}>-\infty$ and with $\theta\leqslant 0$ only if $\beta_{s_i}<\infty$.

It follows from (3.24) that

$$\Delta_{s_\lambda}(\theta)=\begin{cases}0, & \lambda\neq i \quad (\lambda=1,\ 2,\ \ldots,\ m),\\ \theta, & \lambda=i.\end{cases} \tag{3.27}$$

The vector $Y(\theta)$ for θ satisfying condition (3.25) $(\theta\geqslant 0)$ or condition (3.26) $(\theta\leqslant 0)$ therefore meets the requirements of inequalities (3.7). For $j\in I_Y$ this follows from (3.27) and from the preceding assumptions on the boundary values of the variable x_{s_i}. For $j\notin I_Y$ this proposition follows from the definition of θ_0 and from the validity of relationship (3.7) for kernel Y.

The vector $Y(\theta)$ is, thus, the kernel of some program of the dual problem.

We now compute the value of the linear form (3.4) corresponding to the kernel $Y(\theta)$.

Applying formula (3.11), we have

$$\tilde{L}(\theta)=\tilde{L}(Y(\theta))=\sum_{\lambda=1}^{m} b_\lambda y_\lambda(\theta)-\sum_{j=1}^{n}\gamma_j(\theta)\,\Delta_j(\theta),$$

where

$$\gamma_j(\theta)=\begin{cases}\alpha_j \text{ for } \Delta_j(\theta)\geqslant 0,\\ \beta_j \text{ for } \Delta_j(\theta)<0.\end{cases}$$

Since $\Delta_j(\theta)\,\Delta_j\geqslant 0$ for $j\notin I_Y$, $\Delta_{s_\lambda}(\theta)=\Delta_{s_\lambda}=0$ for $\lambda\neq i$, and $\Delta_{s_i}(\theta)=\theta$ we have

$$\gamma_j(\theta)=\begin{cases}\gamma_j, & \text{if } j\neq s_i,\\ \alpha_{s_i}(\beta_{s_i}), & \text{if } j=s_i \text{ and } \theta\geqslant 0\,(\theta<0).\end{cases} \tag{3.28}$$

Here the γ_j refer to the kernel Y.

We transform the expression for $\tilde{L}(\theta)$ applying (3.21), (3.23), (3.24), and (3.28):

$$\tilde{L}(\theta)=\sum_{\lambda=1}^{m} b_\lambda y_\lambda+\theta\sum_{\lambda=1}^{m} b_\lambda e_{i\lambda}-\sum_{j\notin I_Y}\gamma_j\Delta_j(\theta)-\theta\gamma_{s_i}=$$
$$=\sum_{\lambda=1}^{m} b_\lambda y_\lambda-\sum_{j\notin I_Y}\gamma_j\Delta_j+\theta\,[\sum_{\lambda=1}^{m}(b_\lambda-\sum_{j\notin I_Y}\gamma_j a_{\lambda j})\,e_{i\lambda}]-\gamma_{s_i}\theta.$$

Using our notations,

$$b_\lambda-\sum_{j\notin I_Y}\gamma_j a_{\lambda j}=b_\lambda-\sum_{j\notin I_Y}x_j a_{\lambda j}=a_{\lambda 0},$$

therefore

$$\tilde{L}(\theta)=\sum_{\lambda=1}^{m} b_\lambda y_\lambda-\sum_{j\notin I_Y}\gamma_j\Delta_j+\theta\,[\sum_{i=1}^{m} a_{\lambda 0}\,e_{i\lambda}-\gamma_{s_i}]=\tilde{L}(0)+\theta\,(x_{i_0}-\gamma_{s_i}).$$

Thus,

$$\tilde{L}(Y(\theta))=\tilde{L}(Y)+\theta\,(x_{i_0}-\gamma_{s_i}). \tag{3.29}$$

We recall that according to (3.28)

$$\gamma_{s_i}=\begin{cases}\alpha_{s_i}, & \text{if } \theta>0 \text{ (case (i))},\\ \beta_{s_i}, & \text{if } \theta<0 \text{ (case (ii))}.\end{cases}$$

3-6. We will now investigate cases (b) and (c) given in 3-4.

1. We consider the two possibilities terminating in case (b).

For some $i=r$, $x_{r0}<\alpha_{s_i}$ and x_{rj} satisfy conditions (3.19). We apply to kernel Y the elementary transformation associated with the vector A_{s_r} and

$\theta>0$ (case (i)). Since $a_{s_r}>x_{r_0}>-\infty$, this transformation produces the kernel $Y(\theta)$ for any θ, $0\leq\theta<\theta_0$. Restraints (3.19) indicate that for any $j\notin I_Y$ either $x_{rj}=0$ or $\frac{\Delta_j}{x_{rj}}\geq 0$. Hence, $\theta_0=\infty$. $Y(\theta)$ is thus the kernel of some feasible program of the dual probelm for any nonnegative θ.

Applying formula (3.29) (for $i=r$) and since $\gamma_{s_r}=a_{s_r}$ we have

$$\tilde{L}(Y(\theta))=\tilde{L}(Y)+\theta(x_{r_0}-a_{s_r}).$$

By assumption, $x_{r_0}-a_{s_r}<0$. Hence,

$$\lim_{\theta\to\infty}\tilde{L}(Y(\theta))=-\infty,$$

i.e., the linear form of the dual problem is unbounded below in the set of feasible programs. According to Chapter 3, Lemma 1.3, this implies that the restraints of the primal problem are inconsistent.

The second possibility terminating in case (b), when for some $i=r$ $x_{r_0}>\beta_{s_r}$ and conditions (3.20) are satisfied, is analyzed analogously. Here we apply the elementary transformation associated with the vector A_{s_r} and $\theta<0$. It can easily be seen that in this case $\theta_0=\infty$ and, consequently, the restraints of the primal problem are inconsistent.

Case (b) thus amounts to unsolvability of problem (3.1)–(3.3), as the result of inconsistent restraints (3.2) and (3.3).

2. Now consider case (c). We take the index r (a position in the basis of kernel Y such that

$$\text{i) } x_{r_0}<a_{s_r},$$

or

$$\text{ii) } x_{r_0}>\beta_{s_r}.$$

Consider (i). We apply to kernel Y the elementary transformation associated with the vector A_{s_r} for $\theta>0$. Since case (b) does not obtain, $\theta_0<\infty$. Let $Y'=Y(\theta_0)$, where θ_0 is determined from (3.25) with i substituted for r.

Let θ_0 defined in (3.25) be obtained for $j=k$, i.e.,

$$\theta_0=-\frac{\Delta_k}{x_{rk}}. \tag{3.30}$$

Then

$$\Delta_k(\theta_0)=0.$$

From the definition of elementary transformation it follows that for $\lambda=1, 2, \ldots, r-1, r+1, \ldots, m$, $\Delta_{s_\lambda}(\theta_0)=\Delta_{s_\lambda}=0$. Hence

$$\Delta_j(\theta_0)=\sum_{i=1}^{m}a_{ij}y_i(\theta_0)-c_j=0 \tag{3.31}$$

for $j=s_1, s_2, \ldots, s_{r-1}, k, s_{r+1}, \ldots, s_m$.

Consider the system of restraint vectors

$$A_{s_1}, A_{s_2}, \ldots, A_{s_{r-1}}, A_k, A_{s_{r+1}}, \ldots, A_{s_m}. \tag{3.32}$$

Since $x_{rk}\neq 0$ and the system of vectors $A_{s_1}, \ldots, A_{s_{r-1}}, A_{s_r}, A_{s_{r+1}}, \ldots, A_{s_m}$ constituting the basis of pseudoprogram X is linearly independent, we have, from Chapter 4, Theorem 2.1, that system (3.32) is also linearly independent. Hence, applying inequalities (3.31), we find that Y' is the kernel of a support program of the dual problem. The basis of kernel Y' is obtained from the basis of kernel Y when the vector A_{s_r} is replaced by the vector A_k, where k is determined from (3.30).

Since the problem is assumed to be nondegenerate

$$\Delta_j(\theta_0) \neq 0 \quad \text{for} \quad j \notin I_{Y'}.$$

Hence, equality (3.30) is possible for a unique $j=k$ only. Therefore, in the nondegenerate case, condition (3.30) uniquely defines the index k of the vector to be introduced into the basis.

According to (3.29),

$$\tilde{L}(Y') = \tilde{L}(Y) + \theta_0 (x_{r0} - \alpha_{s_r}). \tag{3.33}$$

As we have already indicated, in the nondegenerate case $\theta_0 > 0$. Hence,

$$\tilde{L}(Y') < \tilde{L}(Y).$$

Thus, for (i) of case (c), we see that the elementary transformation associated with the vector $A_{s_r}(x_{r0} < \alpha_{s_r})$ and $\theta > 0$ gives the kernel Y' of a support program of the dual problem decreasing the linear form (3.4).

Consider the second possibility (ii) when $x_{r0} > \beta_{s_r}$. In this case we apply the elementary transformation associated with the vector A_{s_r} for $\theta < 0$. Proceeding as previously, we show that the vector

$$Y' = Y(-\theta_0)$$

is the kernel of a support program of the dual problem. The basis of the kernel Y' is obtained from the system A_j, $j \in I_Y$ when the vector A_k is substituted for A_{s_r}. In this case the index k of the vector A_k to be introduced into the basis is determined from the condition

$$\frac{\Delta_k}{x_{rk}} = \theta_0, \tag{3.34}$$

where θ_0 is computed from (3.26).

The change in the linear form of the dual problem under transformation from Y to Y' is determined from the equality

$$\tilde{L}(Y') = \tilde{L}(Y) - \theta_0 (x_{r0} - \beta_{s_r}), \tag{3.35}$$

whence, applying the inequalities $\theta_0 > 0$ and $x_{r0} - \beta_{s_r} > 0$, we obtain

$$\tilde{L}(Y') < \tilde{L}(Y).$$

If the conditions of case (c) are fulfilled, we may proceed from the given kernel of a support program of the dual problem to the kernel of another support program decreasing the linear form (3.4).

Let X' be the pseudoprogram of the primal problem associated with the kernel Y' and the basis $I_{Y'}$. Let us consider transformation from pseudoprogram X to pseudoprogram X'. The basis of pseudoprogram X $(A_j, j \in I_Y)$ is obtained from the basis of pseudoprogram X' $(A_j, j \in I_{Y'})$ when the vector A_k is substituted for A_{s_r}. The index k of the vector to be introduced into the basis is determined in (i) from (3.30) and in (ii) from (3.34). The extra-basis components x_j of the pseudoprogram X' for $j \neq s_r$ coincide with the corresponding components of pseudoprogram X, and

$$x_{s_r} = \begin{cases} \alpha_{s_r} \text{ (in (i))}, \\ \beta_{s_r} \text{ (in (ii))}. \end{cases}$$

The basis components of X' can be determined from (3.15) with $I_{Y'}$ substituted for I_Y. Observe that

$$\tilde{L}(Y) = L(X) = \sum_{j=1}^{n} c_j x_j, \tag{3.36}$$

where X is a pseudoprogram of the primal problem associated with kernel Y of a support program of problem (3.4)–(3.6). The proof of equality (3.36) is left to the reader (see Exercise 7).

3-7. The dual simplex method when applied to a problem with bilateral restraints involves examining pseudoprograms of the primal problem.

The method consists of identical iterations. Each iteration is subdivided into two stages. In the first stage the conditions specifying cases (a) and (b) are checked. If one of these cases obtains, the process of solution is terminated. If case (c) obtains, we proceed with the second stage of the iteration, in which a new pseudoprogram decreasing linear form (3.1) is constructed.

Successive iterations are continued until a solution is obtained (case (a)) or unsolvability is established (case (b)).

In the nondegenerate case each iteration decreases linear form (3.1), or equivalently, by virtue of (3.36), it decreases linear form (3.4). There is, therefore, no possibility of returning to a pseudoprogram which has previously been examined. Further, since the number of various pseudoprograms of the primal problem is finite, we say that the method is finite. In the next section we shall establish this point for the degenerate case.

It follows from the preceding description of the dual method, that bilateral restraints have almost no effect on the laboriousness of computations in a single iteration. This will become even more obvious once we have considered the dual-simplex algorithms, discussed in 5-5 and 6-5.

Worth mentioning is an additional advantage of the dual method connected with the determination of the initial pseudoprogram of the primal problem (or the kernel of a support program of the dual problem). Let all the variables x_j of the primal problem be bilaterally restrainted by finite boundary values. Then $I' = I'' = (1, 2, \ldots, n)$. Conditions (3.7), which are necessary and sufficient for the vector $Y = (y_1, y_2, \ldots, y_m)$ to be the kernel of some feasible program of the dual problem, will therefore no longer restrict the choice of Y.

Under these assumptions, any m-dimensional vector $Y = (y_1, y_2, \ldots, y_m)$ is the kernel of a feasible program of problem (3.4)–(3.6). Correspondingly, in this case, any linearly independent system of m restraint vectors can be adopted as the basis of the kernel of a support program of the dual problem or of a pseudoprogram of the primal problem.

Thus, if all the variables have finite bilateral restraints, the choice of the initial pseudoprogram is reduced to the choice of an arbitrary linearly independent system comprising m restraint vectors A_j.

While in the simplex method bilateral restraints complicated the process by which the first approximation was determined, in the dual simplex method bilateral restraints, conversely, essentially simplify the determination of initial data for the solution process.

We remark further that when determining the elementary transformation, we took θ_0 to correspond to the first zero of $\Delta_j(\theta)$ for $j \notin I_Y$. This rule induces motion over adjoining vertices of the polyhedral restraint set of the dual problem. It may, however, turn out that when θ is further increased the vector $Y(\theta)$ remains the kernel of a program of problem (3.4)–(3.6) and linear form (3.4) keeps decreasing. It is then convenient to exceed the limit θ_0 but the computations in a single iteration become in this case somewhat more involved. However, the number of iterations is now reduced since several vertices at a time are crossed on the polyhedral restraint

set of the dual problem. This approach is closely connected with the so-called piecewise-linear problems /130, 31/.

3-8. In conclusion we shall briefly review the sequence of operations constituting a single iteration of the dual method.

Before starting with a current iteration, we have a pseudoprogram X with the basis $(A_{s_1}, \dots, A_{s_m})$ and the corresponding parameters x_{ij} and Δ_j. The first stage of the iteration starts with testing pseudoprogram X for optimality. If all the basis components of X satisfy restraints (3.3), we have case (a) implying that X is an optimal program of the primal problem.

When some of the basis components exceed the corresponding boundary values, the parameters x_{ij} and Δ_j should be used to check the conditions corresponding to case (b) (see formulas (3.19) and (3.20)). If these conditions are satisfied, we conclude that the problem is unsolvable. Otherwise, we proceed with the second stage of the iteration.

Among the basis vectors of pseudoprogram X corresponding to the basis components which exceed their boundary values we choose an arbitrary vector A_{s_r}. This vector is to be eliminated from the basis. The choice of the vector A_k to be substituted for A_{s_r} is made following different rules depending on the boundary that the basis component x_{r_0} has actually exceeded. If $x_{r_0} < \alpha_{s_r}$, the index k is defined by (3.30), and θ_0 is computed from (3.25). If $x_{r_0} > \beta_{s_r}$, the required index is obtained from (3.34), where θ_0 is defined by (3.26).

The new pseudoprogram X' and the corresponding parameters x'_{ij} and Δ'_j can be computed from previous data by recurrence formulas (see 5-5).

Two different numerical techniques of the dual method are considered in 5-5 and 6-5.

§ 4. Degeneracy

4-1. Until now we presented the dual simplex method assuming nondegeneracy of the dual problem. In the present section we shall not make this restriction. First let us mention the difficulties which may arise when the dual method is applied to the degenerate case. For the sake of simplicity we shall limit the discussion to a linear-programming problem in canonical form (1.1)–(1.3).

Let $Y=(y_1, y_2, \dots, y_m)$ be an arbitrary support program of the dual problem (1.4)–(1.5) with the basis $A_{s_1}, A_{s_2}, \dots, A_{s_m}$. The basis of a nondegenerate program Y comprising the vectors A_j, $j \in I_Y = (s_1, s_2, \dots, s_m)$ is uniquely defined. If, however, Y is a degenerate program, i. e., some of the

$$\Delta_j = \sum_{i=1}^{m} a_{ij} y_i - c_j$$

for $j \notin I_Y$ are zero, there may be several bases. It can easily be shown that the number of these bases is at most $C^{\gamma}_{m+\gamma}$ where γ is the number of indices $j \notin I_Y$ for which $\Delta_j = 0$. There exist problems for which this number of bases is actually obtained (see Exercise 8). According to the definition of a pseudoprogram, to each basis of a support program Y of the dual problem there corresponds a pseudoprogram of the primal problem with the same basis. In general, to a degenerate support program of the dual problem there correspond several pseudoprograms of the primal problem.

We will now review the various operations involved in a single iteration of the dual method mentioning in particular where the assumption of nondegneracy is used.

In the first stage this assumption is not used. In the second stage nondegeneracy of the dual problem is assumed twice:

1) when computing θ_0,

2) when choosing the vector A_k to be introduced into the basis.

We shall now dwell briefly on each of the above.

1. The parameter θ_0 is computed from (1.23):

$$\theta_0 = \min_{x_{rj}<0}\left(-\frac{\Delta_j}{x_{ij}}\right).$$

When the support program Y is degenerate, it may prove that for some $j \notin I_Y$

$$x_{rj} < 0, \text{ and } \Delta_j = 0,$$

and, consequently, $\theta_0 = 0$. If $\theta_0 = 0$ the new support program Y' coincides with the previous program Y. The iteration, thus, merely produces an alternative basis of program Y. The linear form of the dual problem, naturally, retains its previous value.

In terms of the primal problem, the iteration in question transforms pseudoprogram X into pseudoprogram X'. Pseudoprograms X and X' are associated with two different bases of a given support program Y of the dual problem. The linear form has, therefore, the same value on these pseudoprograms.

If θ_0 remains equal to zero during several iterations, the process of solution throughout these iterations involves transformations of pseudoprograms of the primal problem corresponding to one given support program of the dual problem. The linear form retains a constant value throughout, and the possibility of returning to a pseudoprogram which has previously been examined is not excluded, i.e., cycling may occur.

An example of a linear-programming problem which leads to cycling when solved by the dual simplex method was first constructed by Beale. We shall not dwell here in detail on the occurrence and nature of cycling in the dual method, as we did for the simplex method in Chapter 5, § 9.

We have already noted that the solution of a primal problem by the dual simplex method is equivalent to the application of the simplex method to the dual problem (for details, see § 8). Therefore, to each example of a cycle in the simplex method there corresponds an example of a cycle in the dual simplex method, and vice versa.

Indeed, suppose that cycling is detected in the solution of some problem (problem A) by the simplex method. Consider the linear programming problem (problem B) which is dual with respect to problem A. We reduce problem B to canonical form and apply the dual simplex method.

Pseudoprograms of problem B which correspond to the support programs of problem A constituting the cycle in question form a dual-simplex cycle in problem B.

Thus, the discussion on cycling in the simplex method (Chapter 5, § 9) can be extended to cycling in the dual simplex method. Hence, cycling may arise in the solution process of linear-programming problems by the dual simplex method; the minimum length of a cycle involves six iterations.

Thus, in the degenerate case, θ_0 sometimes vanishes, which may result in cycling.

2. The vector A_k to be introduced into the basis is chosen with the help of the formula

$$-\frac{\Delta_k}{x_{rk}} = \theta_0 = \min_{x_{rj}<0}\left(-\frac{\Delta_j}{x_{rj}}\right). \tag{4.1}$$

We have already indicated that if the program Y' of the dual problem obtained under elementary transformation of program Y with $\theta = \theta_0$ is nondegenerate, the index k is uniquely defined by (4.1). If, however, Y' is a degenerate program, i.e.,

$$\Delta'_j = \Delta_j(\theta_0) = \Delta_j + \theta_0 x_{rj} = 0$$

for some $j \in I_Y$, relationship (4.1), generally speaking, is not sufficient to define the vector to be introduced into the basis uniquely.

Taking A_k as an arbitrary restraint vector from among the several vectors satisfying relationship (4.1), may lead to cycling. If, however, the rule for choosing the vector is somewhat improved (extended), this vector can be uniquely defined and cycling need not be anticipated.

4-2. Before deriving the extended rule, we shall clarify the geometrical meaning of degeneracy in terms of the dual simplex method. As always, we start with the first geometrical interpretation.

Consider pseudoprogram X corresponding to a support program of the dual problem. Let the system of restraint vectors A_j with $j \in I_Y = (s_1, s_2, \ldots, s_m)$ constitute the basis of program Y, and also the basis of pseudoprogram X. According to (2.2), the polyhedral cone K_X introduced in 2-1 lies in the lower halfspace with respect to the linear-form hyperplane (1.1) (Π_X) passing through the point X. Again, according to (2.2) the condition $\Delta_l = 0$ $(l \notin I_Y)$ geometrically means that the edge of cone K_X formed by the intersection of hyperplanes (1.2) and $x_j = 0$ for $j \notin I_Y$, $j \neq l$, is contained in hyperplane Π_X. Therefore, geometrically, nondegeneracy of support program Y of the dual problem indicates that the cone K_X and the hyperplane Π_X have only one point in common, the point X.

Now let the support program Y be degenerate. Consider the polyhedral set M_X comprising the intersection of the cone K_X and the halfspace $x_{s_r} \leq 0$ (the vector A_{s_r} is to be eliminated from the basis). Since

in the degenerate case some of the edges of the cone K_X belong to the hyperplane Π_X, some of the vertices of the polyhedral set M_X (except the vertex X, which is invariably contained in Π_X) will also belong to hyperplane Π_X. These vertices are, obviously, pseudoprograms of the primal problem on which the linear form (1.1) has the same value.

In 2-2 we showed that the geometrical image of pseudoprogram X' (the pseudoprogram improved in an iteration) is the vertex of polyhedral set M_X which is closest to hyperplane Π_X. In this case the polyhedral set M_X has vertices (other than X) which lie on hyperplane Π_X. By a successive iteration we will transfer from one of these vertices to another. This transformation will, obviously, have no effect on the value of linear form (1.1). If we choose the vertex X' on hyperplane Π_X without following certain definite rules, we may, after several iterations, return to the initial point X, i.e., a cycle will form.

Let us now give the second geometrical interpretation of the dual simplex method (see 2-7). Here the image of a support program Y of the dual problem is the hyperplane Π spanned by the augmented basis vectors of this program, $\overline{A}_{s_1}, \overline{A}_{s_2}, \ldots, \overline{A}_{s_m}$. Nondegeneracy of program Y indicates that for any $j \notin I_Y$ the augmented restraint vector $\overline{A}_j$ is strictly above the hyperplane Π. In the degenerate case some of the vectors $\overline{A}_j$, $j \notin I_Y$ may belong to hyperplane Π. We have indicated in 2-7 that, geometrically, each iteration involves rotation of hyperplane Π about the vectors $\overline{A}_{s_1}, \overline{A}_{s_2}, \ldots, \overline{A}_{s_{r-1}}, \overline{A}_{s_{r+1}}, \ldots, \overline{A}_{s_{m-1}}, \overline{A}_{s_m}$ until the hyperplane first meets one of the vectors $\overline{A}_j$, where $j \notin I_Y = (s_1, s_2, \ldots, s_m)$. The rotation should be so carried out that the vector $\overline{A}_{s_r}$ remains below the hyperplane. The vector $\overline{A}_k$ contained in the new hyperplane Π' is substituted for the vector $\overline{A}_{s_r}$ in the augmented basis of the new support program Y'. Hyperplane Π' is the geometrical image of the new support program Y'. It is geometrically obvious that if Y is a nondegenerate program, hyperplane Π' is different from hyperplane Π, i.e., $Y' \neq Y$. In the degenerate case it may occur that any infinitesimal rotation entailed in a given iteration will bring the vector $\overline{A}_k$ $(k \notin I_Y)$ to above the new hyperplane. In this case the new support program Y' coincides with the initial program Y. The iteration thus merely produces an alternative basis of program Y.

The geometrical interpretation of degeneracy not only offers a more complete understanding of this phenomenon, but also hints at the means of avoiding the dangers of degeneracy, i.e., the possibility of cycling. This is quite obvious in view of the first geometrical interpretation. We have seen that nondegeneracy of the dual problem (1.4)–(1.5), in terms of the first geometrical interpretation, indicates that any hyperplane of the form

$$(C, X) = \text{const}$$

meets at most one pseudovertex of the polyhedral restraint set of problem (1.1)–(1.3). It is geometrically obvious that this requirement can always be satisfied by a sufficiently small modification of the coefficients c_j in linear form (1.1). The above is the basis from which the rule eliminating all possibility of cycling as the result of injudicious choice of the vector to be introduced into the basis is derived.

4-3. We will now establish the rule for choosing the vector to be introduced into the basis. Since the linear-programming problem in canonical form is a particular case of a problem with bilateral restraints, we shall give our arguments with reference to problem (3.1)–(3.3).

Let ε be a positive number.

Consider the following problem with bilateral restraints.

Maximize the linear form

$$\sum_{j=1}^{n} c_j(\varepsilon) x_j \tag{4.2}$$

subject to the conditions

$$\sum_{j=1}^{n} A_j x_j = B, \tag{4.3}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j. \tag{4.4}$$

Here

$$c_j(\varepsilon) = c_j + (-1)^{\nu_j} \varepsilon^{n+1-j}, \quad j = 1, 2, \ldots, n, \tag{4.5}$$

$$\nu_j = \begin{cases} 1, & \text{if } 1 \leqslant j \leqslant m, \text{ or } x_j^{(0)} = \alpha_j, \\ 0, & \text{if } x_j^{(0)} = \beta_j, \ j \geqslant m+1. \end{cases} \tag{4.6}$$

The $x_j^{(0)}$ appearing in formula (4.6) constitute the pseudoprogram $X_0 = (x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)})$ of problem (3.1)–(3.3) with the basis $A_1, A_2, \ldots, A_m$.

The preceding problem differs from problem (3.1)–(3.3) only in the linear-form coefficients which are

obtained from the corresponding c_j with the aid of formula (4.5) and the parameter ε. Problems of the form (4.2)–(4.4) will, henceforth, be called ε-problems corresponding to problem (3.1)–(3.3). The rule for choosing the vector to be introduced into the basis is based on three propositions concerning ε-problems.

Theorem 4.1. There exists $\varepsilon_1 > 0$ such that if the vector $X_0=(x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)})$ is a pseudoprogram of problem (3.1)–(3.3) with the basis $A_1, A_2, \ldots, A_m$, this vector is also a pseudoprogram of the ε-problem (4.2)–(4.4) for any positive $\varepsilon < \varepsilon_1$.

Proof. Let

$$A_j=\sum_{i=1}^{m} x_{ij}^{(0)} A_i, \qquad j=1, 2, \ldots, n.$$

In this case the parameters $\Delta_j(\varepsilon)$ corresponding to ε-problem (4.2)–(4.4) and to the system of vectors $A_1, A_2, \ldots, A_m$ have the form

$$\Delta_j(\varepsilon)=\sum_{i=1}^{m} c_i(\varepsilon)\, x_{ij}^{(0)}-c_j(\varepsilon)=\Delta_j-(-1)^{\nu_j}\varepsilon^{n+1-j}+\sum_{i=1}^{m}(-1)^{\nu_i}\varepsilon^{n+1-i}x_{ij}^{(0)}. \tag{4.7}$$

Let

$$\zeta=\max_{m+1\leqslant j\leqslant n}\sum_{i=1}^{m}|x_{ij}^{(0)}|.$$

We choose ε_1 as follows

$$\varepsilon_1=\min\left\{1, \frac{1}{\zeta}\right\}. \tag{4.8}$$

Then, for $\varepsilon < \varepsilon_1$

$$\left|\sum_{i=1}^{m}(-1)^{\nu_j}\varepsilon^{n+1-i}x_{ij}^{(0)}\right|\leqslant\varepsilon^{n+1-m}\sum_{i=1}^{m}\varepsilon^{m-i}|x_{ij}^{(0)}|\leqslant\varepsilon^{n+1-m}\zeta<\varepsilon^{n-m}. \tag{4.9}$$

These inequalities follow directly from (4.8) defining ε_1.

Applying equality (4.7) (for $j\geqslant m+1$) and inequality (4.9), we obtain

$$\Delta_j(\varepsilon)\begin{cases} >\Delta_j, & \text{if } \nu_j=1, \\ <\Delta_j, & \text{if } \nu_j=0. \end{cases} \tag{4.10}$$

Since the vector X_0 is a pseudoprogram of problem (3.1)–(3.3) with the basis $A_1, A_2, \ldots, A_m$, then

$$\Delta_j\begin{cases} \geqslant 0, & \text{if } x_j^0=\alpha_j, \\ \leqslant 0, & \text{if } x_j^{(0)}=\beta_j \end{cases}$$

(see (3.14)).

Hence, applying conditions (4.6) and (4.10) we obtain

$$\Delta_j(\varepsilon)\begin{cases} >\Delta_j\geqslant 0, & \text{if } x_j^{(0)}=\alpha_j, \\ <\Delta_j\leqslant 0, & \text{if } x_j^{(0)}=\beta_j, \end{cases} \tag{4.11}$$
$$m+1\leqslant j\leqslant n.$$

According to (3.14) inequalities (4.11) indicate that the vector X_0 is a pseudoprogram of the ε-problem (4.2)–(4.4). This completes the proof.

Theorem 4.2. There exists $\varepsilon_2 > 0$ such that for $0<\varepsilon<\varepsilon_2$ the dual problem of problem (4.2)–(4.4) is nondegenerate.

Proof. Let $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$ be a linearly independent system of restraint vectors. The set of indices of these vectors is denoted by I. The other restraint vectors A_j are then expanded in terms of the vectors of this system

$$A_j=\sum_{i=1}^{m} x_{ij}^{(I)}A_{s_i} \tag{4.12}$$

and we take

$$\Delta_j^{(I)}(\varepsilon)=\sum_{i=1}^{m} c_{s_i}(\varepsilon)\, x_{ij}^{(I)}-c_j(\varepsilon). \tag{4.13}$$

Applying (4.5), we see that $\Delta_i^{(I)}(\varepsilon)$ for any $j\notin I$ is a polynomial of order not higher than n. The polynomial $\Delta_j^{(I)}(\varepsilon)\neq$ const, since it contains ε^{n+1-j}. Hence, this polynomial has at most n roots.

Let the least of the positive roots of the polynomial $\Delta_j^{(I)}(\varepsilon)$ be $\eta_j^{(I)} > 0$. Let

$$\eta^{(I)} = \min_{j \notin I} \eta_j^{(I)} > 0.$$

The parameter $\eta^{(I)} > 0$ is defined by a system of linearly independent vectors A_j, $j \in$ The number of such systems is finite. Therefore

$$\varepsilon_2 = \min_I \eta^{(I)} > 0.$$

Let $0 < \varepsilon < \varepsilon_2$. Consider a pseudoprogram X of problem (4.2)–(4.4) with the basis comprising the vectors A_j, $j \in I_X$. From formulas (4.12) and (4.13), where I_X has been substituted for I, we compute the parameters $\Delta_j^{(I_X)}(\varepsilon)$ of this program.

By assumption, the least positive root of the polynomial $\Delta_j^{(I_X)}(\varepsilon)$, $j \notin I_X$, is on the right of ε_2. Hence

$$\Delta_j^{(I_X)}(\varepsilon) \neq 0 \text{ for } 0 < \varepsilon < \varepsilon_2.$$

The parameters $\Delta_j^{(I_X)}(\varepsilon)$ $(j \notin I_X)$ of any pseudoprogram X of problem (4.2)–(4.4) are thus nonzero. By definition, this implies that the dual problem of problem (4.2)–(4.4) is nondegenerate. This completes the proof.

Theorem 4.3. There exists $\varepsilon_3 > 0$ such that if

$$0 < \varepsilon < \varepsilon_3$$

and X is a pseudoprogram of the ε-problem, this vector is also a pseudoprogram of problem (3.1)–(3.3) with the same basis. Moreover, if X is an optimal program of problem (4.2)–(4.4), it solves the initial problem (3.1)–(3.3).

Proof. Consider a linearly independent system of m vectors A_j, $j \in I$. From formulas (4.12), (4.13) we compute

$$\Delta_j^{(I)} = \Delta_j^{(I)}(0), \; j \notin I.$$

Let

$$\Delta^{(I)} = \min_{\Delta_j^{(I)} \neq 0} |\Delta_j^{(I)}|$$

(if all $\Delta_j^{(I)} = 0$, then $\Delta^{(I)} = \infty$). Let

$$\Delta = \min_I \Delta^{(I)} > 0, \tag{4.14}$$

where the minimum is taken over the various possible linearly independent systems A_j, $j \in I$. We further define

$$\sigma = \max_I \left[1 + \sum_{i=1}^{m} |x_{ij}^{(I)}|\right] < \infty, \tag{4.15}$$

where $x_{ij}^{(I)}$ are given by (4.12) and the maximum is again taken over the various possible systems of A_j, $j \in I$.

The ε_3 specified in the theorem can now be computed from

$$\varepsilon_3 = \min\left\{1, \frac{\Delta}{\sigma}\right\} > 0, \tag{4.16}$$

where Δ and σ are defined in (4.14) and (4.15), respectively.

We shall show that ε_3 defined in (4.16) indeed satisfies the requirements of the theorem. To this end, taking $0 < \varepsilon < \varepsilon_3$, we consider a pseudoprogram X of problem (4.2)–(4.4).

Let the basis of pseudoprogram X comprise the vectors A_j, $j \in I_X = (s_1, s_2, \ldots, s_m)$. By assumption,

$$\Delta_j^{(I_X)}(\varepsilon) \begin{cases} \geqslant 0 & \text{for } x_j = \alpha_j, \\ \leqslant 0 & \text{for } x_j = \beta_j, \end{cases} \; j \notin I_X. \tag{4.17}$$

From (4.7)

$$\Delta_j^{(I_X)}(\varepsilon) = \Delta_j^{(I_X)} - (-1)^{\nu_j} \varepsilon^{n+1-j} + \sum_{i=1}^{m} (-1)^{\nu_{s_i}} \varepsilon^{n+1-s_i} x_{ij}^{(I_X)}. \tag{4.18}$$

It follows from (4.18) and (4.15) that

$$\left|\Delta_j^{(I_X)}(\varepsilon) - \Delta_j^{(I_X)}\right| \leqslant \varepsilon\sigma.$$

Let $\Delta_j < 0$. Then

$$\Delta_j(\varepsilon) \leqslant \Delta_j + \varepsilon\sigma \leqslant -\Delta + \varepsilon\sigma < -\Delta + \varepsilon_3\sigma \leqslant 0.$$

In this chain of relationships, the second inequality follows from (4.14), the third inequality from the assumption concerning ε, and the fourth from the definition of ε_3 (see (4.16)).

Thus, for $\Delta_j < 0$ the parameters $\Delta_j(\varepsilon) < 0$. Analogously we verify that when $\Delta_j > 0$, $\Delta_j(\varepsilon)$ are also positive.

We can now easily show that the vector X is a pseudoprogram of problem (3.1)–(3.3) with the basis $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$.

Indeed,

$$\Delta_j \geqslant 0 \quad \text{for} \quad x_j = \alpha_j,$$

since otherwise we would have obtained $\Delta_j(\varepsilon) < 0$ in contradiction to (4.17). Similarly, for $x_j = \beta_j$

$$\Delta_j \leqslant 0.$$

The vector X, which satisfies conditions (3.2) and (3.14), is, thus, a pseudoprogram of problem (3.1)–(3.3).

To complete the proof of the theorem, it remains to show that the pseudoprogram X which is a feasible program of problem (4.2)–(4.4) (and thus also an optimal program of this problem) solves problem (3.1)–(3.3).

From the above, X is a pseudoprogram of problem (3.1)–(3.3). By assumption, the vector X is also a feasible program of this problem. Hence, according to the optimality test, the vector X solves problem (3.1)–(3.3). This completes the proof.

4-4. We are now in a position to extend the dual simplex method to any problem with bilateral restraints, without the corresponding dual problem becoming nondegenerate.

Rather than consider problem (3.1)–(3.3), we turn to the corresponding ε-problem (4.2)–(4.4) with

$$0 < \varepsilon < \varepsilon_0 = \min\{\varepsilon_1, \varepsilon_2, \varepsilon_3\}.$$

The positive numbers ε_1, ε_2, ε_3 are specified in Theorems 4.1, 4.2, and 4.3, respectively.

Let X_0 be a pseudoprogram of problem (3.1)–(3.3) with the basis $A_1, A_2, \ldots, A_m$. According to Theorem 4.1 ($\varepsilon < \varepsilon_1$) the vector X_0 is also a pseudoprogram of the ε-problem with the same basis.

Starting with pseudoprogram X_0 we proceed to solve the ε-problem (4.2)–(4.4). By virtue of Theorem 4.2, this problem is nondegenerate ($\varepsilon < \varepsilon_2$). Therefore, after a finite number of iterations, having examined pseudoprograms $X_1, \ldots, X_{N-1}$, we arrive at solution X_N of problem (4.2)–(4.4).

According to Theorem 4.3 ($\varepsilon < \varepsilon_3$) the vectors X_i, $1 \leqslant i \leqslant N$, are pseudoprograms of problem (3.1)–(3.3), the last vector in the chain X_N also being a solution of this problem.

Thus, given a sufficiently small ε, the dual simplex method applied to problem (4.2)–(4.4) involves covering various pseudoprograms of the initial problem (3.1)–(3.3) eventually obtaining the required optimal program.

Let us consider this procedure in a single iteration, which involves transformation from pseudoprogram X to pseudoprogram X'. The vector X is a pseudoprogram of both the initial problem (3.1)–(3.3) and the ε-problem (4.2)–(4.4). The vector X should, obviously, be considered as a pseudoprogram of the ε-problem only while choosing the vector A_k to introduced into the new basis.

As in our discussion of the dual method with reference to problems with bilateral restraints, we again distinguish two possibilities:

1. The vector A_{s_r} is eliminated from the basis, where

$$x_{r0} < \alpha_{s_r}.$$

In this case the index k of the vector A_k to be introduced into the new basis is determined from

$$-\frac{\Delta_k(\varepsilon)}{x_{rk}} = \min_{j \in E}\left(-\frac{\Delta_j(\varepsilon)}{x_{rj}}\right). \tag{4.19}$$

Here E denotes the set of indices j such that

$$-\frac{\Delta_j(\varepsilon)}{x_{rj}} > 0, \; x_{rj} \neq 0,$$

or, equivalently, such that

$$x_{rj}\begin{cases} < 0 & \text{for} \quad x_j = \alpha_j, \quad j \notin I_X, \\ > 0 & \text{for} \quad x_j = \beta_j, \quad j \notin I_X. \end{cases}$$

The basis of pseudoprogram X comprises the vectors $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$. Therefore, following formula

(4.7), we transorm (4.19) to the form

$$-\frac{\Delta_k}{x_{rk}}+\frac{(-1)^{\nu_k}}{x_{rk}}\varepsilon^{n+1-k}-\sum_{l=1}^{m}\frac{(-1)^{\nu_{s_l}}x_{lk}}{x_{rk}}\varepsilon^{n+1-s_l}=$$
$$=\min_{j\in E}\left(-\frac{\Delta_j}{x_{rj}}+\frac{(-1)^{\nu_j}}{x_{rj}}\varepsilon^{n+1-j}-\sum_{l=1}^{m}\frac{(-1)^{\nu_{s_l}}x_{lj}}{x_{rj}}\varepsilon^{n+1-s_l}\right). \tag{4.20}$$

Since ε is an arbitrarily small positive number, the search for the minimum in (4.20) and for the index on which the minimum is obtained involves comparing, successively, the coefficients preceding the various powers of ε, i.e., $\varepsilon^0=1$, ε, ε^2, ..., ε^n. From this we obtain the following rule for the determination of k.

Let the set E_1 comprise all those indices j on which

$$\theta_0=\min_{j\in E}\left(-\frac{\Delta_j}{x_{rj}}\right)$$

is obtained (comparison of the coefficients of $\varepsilon^0=1$). If E_1 contains only one index, it is taken as the required k. Otherwise, the search for index k is resumed. To avoid complicated notations, we take

$$s_1<s_2<\ldots<s_m.$$

This can always be achieved by suitably renumbering the positions of the basis.

Let ε_1 be the set of indices $j\in E_1$ greater than s_m. In ε_1 we define subset $\bar{\varepsilon}_1$ comprising the indices $j\in\varepsilon_1$ such that

$$\frac{(-1)^{\nu_j}}{x_{rj}}<0.$$

If $\bar{\varepsilon}_1$ consists of only one index, then

$$k=\max_{j\in\varepsilon_1} j.$$

If $\bar{\varepsilon}_1$ is an empty set, and $E_1=\varepsilon_1$, then

$$k=\min_{j\in\varepsilon_1=E_1} j.$$

If neither of these conditions is satisfied, i.e., $\bar{\varepsilon}_1$ is an empty set and $E_1\neq\varepsilon_1$, we form the set E_2. This set comprises the indices $j\in E_1$ on which

$$\min_{j\in E_1-\varepsilon_1}\frac{(-1)^{\nu_{s_m}+1}x_{mj}}{x_{rj}}$$

is obtained. Obviously, if ε_1 is an empty set, no intermediate operations need intervene in the transformation from E_1 to E_2.

Observe that the rule of transformation from E_1 to E_2 is based on the comparison of the coefficients of ε^{n+1-j} for $j\geqslant s_m\,(j\in E_1)$. If the set E_2 contains only one index, it is taken as k. Otherwise, transformation from E_2 to E_3 is tried, following the same rules as the transformation from E_1 to E_2. The only difference is that the index s_{m-1} is substituted for s_m. If the set E_3 is insufficient to define the index k, E_3 is transformed to E_4. Successive transformations either produce the index k or exhaust all the sets E_l, which are, obviously, $m+1$ in number.

If the last set E_{m+1} contains several elements, the index k is defined as follows.

We isolate the set $\bar{\varepsilon}_1$ comprising those indices $j\in E_{m+1}$ for which

$$\frac{(-1)^{\nu_j}}{x_{rj}}<0.$$

If $\bar{\varepsilon}_1$ contains at least one element, then

$$k=\max_{j\in\bar{\varepsilon}_1} j.$$

Otherwise,

$$k=\min_{j\in E_{m+1}} j.$$

This last step in the process of determining index k involves comparison of the coefficients of ε^{n+1-j} for $j<s_1$.

The index of the vector A_k to be introduced into the basis is thus established in at most $m+2$ steps.

2. We now proceed with the second case, when

$$x_{r0} > \beta_{s_r}.$$

Here the index k of the vector A_k to be introduced into the new basis is defined from

$$\frac{\Delta_k(\varepsilon)}{x_{rk}} = \min_{j \in E} \frac{\Delta_j(\varepsilon)}{x_{rj}}, \tag{4.21}$$

where E comprises the indices j such that

$$x_{rj} \begin{cases} > 0 & \text{for} \quad x_j = \alpha_j, \quad j \notin I_X, \\ < 0 & \text{for} \quad x_j = \beta_j, \quad j \notin I_X. \end{cases}$$

From (4.21) and (4.7) we see that, for sufficiently small positive ε,

$$\theta_0(\varepsilon) = \min \left(\frac{\Delta_j}{x_{rj}} - \frac{(-1)^{\nu_j}}{x_{rj}} \varepsilon^{n+1-j} + \sum_{i=1}^{m} \frac{(-1)^{\nu_{s_i}} x_{ij}}{x_{rj}} \varepsilon^{n+1-s_i} \right) \tag{4.22}$$

is obtained on the index k.

From (4.22) we easily derive a rule for determining k, which is analogous to what has been described for the first case. Since the expressions being minimized in (4.20) and (4.22) differ only in sign, we shall look for the index k applying the previous rule where x_{rj} has been replaced by

$$\bar{x}_{rj} = -x_{rj}.$$

Our analysis shows that the solution process of the ε-problem (4.2)-(4.4) with sufficiently small ε is equivalent to the solution process of problem (3.1)-(3.3) employing the preceding rule for the determination of the vector to be introduced into the new basis. Application of this rule therefore guarantees that there will be a finite number of steps in the dual method, regardless of the properties of the problem being solved.

4-5. To conclude this section, we return to problem (1.1)-(1.3), written in canonical form. In this case $\alpha_j = 0$, $\beta_j = \infty$ for all $j = 1, 2, \ldots, n$. The second possibility is, therefore, unfeasible, and the set E comprises the indices j such that $x_{rj} < 0$. We also observe that in this case $\nu_j = 1$ for all $j = 1, 2, \ldots, n$. Because of these features of problem (1.1)-(1.3) the rule for determining the index k takes on the following form.

Let E_1 denote the set of indices j on which

$$\theta_0 = \min_{x_{rj} < 0} \left(-\frac{\Delta_j}{x_{rj}} \right)$$

is obtained.

If the index k has not yet been defined (E_1 contains several elements), we proceed with the next step. If all the indices $j \in E_1$ are greater than s_m, then

$$k = \min_{j \in E_1} j.$$

Otherwise, we form the set E_2 comprising all the indices j on which

$$\min_{\substack{j \in E_1 \\ j < s_m}} \frac{x_{mj}}{x_{rj}}$$

is obtained. To elucidate this part of the rule, it suffices to observe that in case of problem (1.1)-(1.3) the set $\bar{e}_i$ always remains empty.

If the set E_2 contains a single element, it is taken as the index k. Otherwise, we proceed with the next step, which either produces the index k or leads to a set of indices E_3 containing more than one element. The rule for transformation to E_3 coincides with the preceding rule, the only difference being that s_{m-1} is substituted for s_m, and $x_{m-1,j}$ for x_{mj}. We recall that according to an earlier assumption $s_1 < s_2 < \ldots \ldots < s_{m-1} < s_m$. The successive steps are repeated until either the index k is determined or the set E_{m+1} is obtained. In the latter case, the index k is defined as the least of $j \in E_{m+1}$:

$$k = \min_{j \in E_{m+1}} j.$$

The preceding rule for determining the index k of the vector A_k to be introduced into the basis eliminates all possibility of returning to a pseudoprogram which has previously been examined. The rule thus fully guarantees against cycling.

§ 5. The first algorithm

5-1. The dual simplex method, like the simplex method, can be realized by two different computational procedures. The difference between the computational procedures (the dual algorithms) lies in the different methods of computation of the parameters Δ_j.

In this section we discuss the computational procedure of the first algorithm with reference to the linear-programming problem in canonical form and to a problem with bilateral restraints. The applications of the algorithm are illustrated by appropriate examples.

When solving a linear-programming problem by the dual simplex method, the initial support program of the dual problem (more precisely, the initial dual basis) is assumed to be known.

Under the first algorithm, the optimum program of a nondegenerate linear-programming problem is computed as follows. The basis of the initial support program of the dual problem is taken as the basis of pseudoprogram X of the primal problem. Expanding the constraint vector in terms of the vectors of the dual basis, we obtain the basis components x_{i0} of pseudoprogram X. The signs of x_{i0} enable us to establish whether the pseudoprogram obtained is a feasible program of the problem. If case (a) obtains, the initial support program of the dual problem is also its solution, and the initial pseudoprogram is the optimal program of the primal problem. If some of the components x_{i0} are negative, we compute the coefficients x_{ij} in the expansion of the restraint vectors in terms of the dual basis. If for some $x_{r0}<0$ all $x_{rj}\geqslant 0$, the problem is unsolvable (case (b)). In this case the process of solution terminates by establishing inconsistency of the problem restraints.

If case (c) obtains, we proceed with the second stage of the iteration. Using formulas (1.11), we compute the parameters Δ_j. Then we determine the index of the vector to be eliminated from the dual basis, and the restraint vector to be introduced in its place. The new dual basis is analyzed in the next iteration in the same way as in the preceding step. The parameters x_{i0}, x_{ij}, and Δ_j are computed from the corresponding parameters of the preceding iteration by recurrence formulas (1.12) from Chapter 5, as in the first algorithm of the simplex method. The computations are continued until the optimal program is obtained or unsolvability of the problem is established. The solution process terminates in a finite number of steps.

We see that the computational procedure of the first algorithm of the dual method is similar to that of the first simplex algorithm. The tableaus in which the iteration parameters are written are also similar in the two methods.

The difference between the methods is that in one case we pass successively from one support program of the problem to another, whereas in the other case from one pseudoprogram to another. In the first case we cover bases of the primal problem, and in the second case bases of the dual problem. The formal difference between the computational procedures lies only in the rules specifying the transformation from a given basis to the next one and in the optimality and unsolvability tests of the programs. In the simplex method, the vector to be introduced into the basis is first determined and then the vector to be eliminated from the basis. In the

dual method, conversely, the vector to be introduced into the basis is chosen after the vector to be eliminated from the basis has been determined. Hence the slight difference in the first algorithm tableaus of the two methods.

TABLE 6.1

l-th tableau

No.	C_X	B_X	A_0	A_1	A_2	...	A_k	...	A_n
1	c_{s_1}	A_{s_1}	$x_{10}^{(l)}$	$x_{11}^{(l)}$	$x_{12}^{(l)}$	...	$x_{1k}^{(l)}$	...	$x_{1n}^{(l)}$
2	c_{s_2}	A_{s_2}	$x_{20}^{(l)}$	$x_{21}^{(l)}$	$x_{22}^{(l)}$	...	$x_{2k}^{(l)}$	...	$x_{2n}^{(l)}$
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	...	⋮
r	c_{s_r}	A_{s_r}	$x_{r0}^{(l)}$	$x_{r1}^{(l)}$	$x_{r2}^{(l)}$	...	$x_{rk}^{(l)}$	...	$x_{rn}^{(l)}$
⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	...	⋮
m	c_{s_m}	A_{s_m}	$x_{m0}^{(l)}$	$x_{m1}^{(l)}$	$x_{m2}^{(l)}$	...	$x_{mk}^{(l)}$	...	$x_{mn}^{(l)}$
$m+1$	—	Δ	$L^{(l)}$	$\Delta_1^{(l)}$	$\Delta_2^{(l)}$	...	$\Delta_k^{(l)}$		$\Delta_n^{(l)}$
$m+2$	—	θ	—			...	$\theta_0^{(l)}$	...	

Comparing the l-th tableau of the first dual algorithm (Table 6.1) with the corresponding tableau in Chapter 5, § 2, we see that the only structural difference in the tableaus is that column θ is replaced by row θ. Column θ in simplex tableaus was used to choose the vector to be eliminated from the basis. Row θ in dual tableaus is used to determine the vector to be introduced into the dual basis.

The column B_X in the l-th tableau contains the basis vectors A_{s_i} of pseudoprogram $X^{(l)}$. In column C_X we write the linear-form coefficients of the basis components of pseudoprogram $X^{(l)}$. In columns A_0 and A_j ($j=1, 2, \ldots, n$) we write the corresponding coefficients of the expansion of the constraint vector and the restraint vectors in terms of the basis of pseudoprogram $X^{(l)}$. The elements $x_{i0}^{(l)}$ in column $A_0=B$ are the basis components of pseudoprogram $X^{(l)}$.

The $(m+1)$-th row in the l-th tableau contains the parameters $\Delta_j^{(l)}$ related to the coefficients $x_{ij}^{(l)}$ and c_j by formulas (1.11). According to the position in the l-th tableau, it is expedient to denote the parameter $\Delta_j^{(l)}$, as in the simplex method, by $x_{m+1,j}^{(l)}$. As before,

$$x_{m+1,0}^{(l)}=L(X^{(l)})=L^{(l)}.$$

The columns $A_0, A_1, \ldots, A_n$ (positions $1, 2, \ldots, m+1$) constitute the principal part of the l-th tableau. The elements $\theta^{(l)}$ of the $(m+2)$-th row are computed from the formula

$$\theta_j^{(l)}=-\frac{\Delta_j^{(l)}}{x_{rj}^{(l)}}.$$

Here r is the position of the vector to be eliminated from the basis $(x_{r0}<0)$. In row $\theta^{(l)}$ only those positions j, for which $x_{rj}<0$, are filled. The other positions are crossed out.

The l-th iteration fills the principal part of the l-th tableau. In the first stage of the $(l+1)$-th iteration we determine which of the three cases obtains. In case (c), we proceed with the second stage of the iteration. In the second stage, we establish the position r of the vector to be eliminated from the basis, compute the row $\theta^{(l)}$, determine the vector A_k to be introduced into the basis, and fill the principal part of the $(l+1)$-th tableau. The elements of the principal part of the $(l+1)$-th tableau are computed from those of the principal part of the preceding tableau by the recurrence formulas

$$x_{ij}^{(l+1)}=\begin{cases} x_{ij}^{(l)}-\dfrac{x_{rj}^{(l)}}{x_{rk}^{(l)}}\,x_{ik}^{(l)} & \text{for } i\neq r,\\[2mm] \dfrac{x_{rj}^{(l)}}{x_{rk}^{(l)}} & \text{for } i=r,\end{cases} \tag{5.1}$$

$$i=1,\ 2,\ \ldots,\ m,\ m+1;\quad j=0,\ 1,\ 2,\ \ldots,\ n.$$

The vector A_{s_r} with the maximum (in absolute value) negative basis component x_{s_r} is eliminated from the basis. The vector A_k to be introduced into the basis is that on which

$$\theta_0^{(l)}=\min\theta_j^{(l)}=\min_{x_{rj}^{(l)}<0}\left(-\frac{\Delta_j^{(l)}}{x_{rj}^{(l)}}\right)$$

is obtained. In nondegenerate problems A_k is uniquely determined from this condition. The rule for choosing the vector A_k in the degenerate case will be discussed in 5-2. The basis position occupied by the vector introduced into the basis in the preceding iteration and the vector to be replaced are marked by arrows. The element θ_0 of row θ is enclosed in a frame. The direction row (r) and the direction column (k) are similarly singled out.

The $(l+1)$-th tableau is transformed to the $(l+2)$-th tableau in the same way as the l-th tableau to the $(l+1)$-th tableau.

The computations are checked as in the first simplex algorithm. At definite intervals, the parameters $\Delta_j=x_{m+1,j}$ are computed not only from the recurrence formulas, but also directly from (1.11).

The bulkness of computations in each iteration in the first dual algorithm is approximately the same as in the first simplex algorithm. The number of multiplications is the same in both cases (the principal parts of the tableaus are transformed according to the same recurrence formulas). Division occurs only in the computation of n and in the transformation of the direction row. In the simplex method the number of divisions is at most n. In the dual method, the number of divisions is at most $2(n-m)$.

The above data, unfortunately, do not hint at the comparative difficulty of solving the general linear-programming problem by either of these methods. At present there is no statistical, and a fortiori no theoretical information for comparing the number of iterations required for solving any given problem by the simplex method and by the dual method.

Figure 6.5 gives a block diagram of the solution of a linear-programming problem by the first dual algorithm.

5-2. In our description of the algorithm, the problem was assumed to be nondegenerate. This assumption ensures a unique choice of the vector to be introduced into the basis.

In 4-3 we stated and proved a rule which guarantees against cycling in the process of solution of a problem by the dual method. Let us consider the application of this rule to the first algorithm.

The restraint vectors are numbered so that the initial basis comprises the vectors $A_1, A_2, \ldots, A_m$. This numeration of the restraint vectors A_j is retained throughout the process of solution.

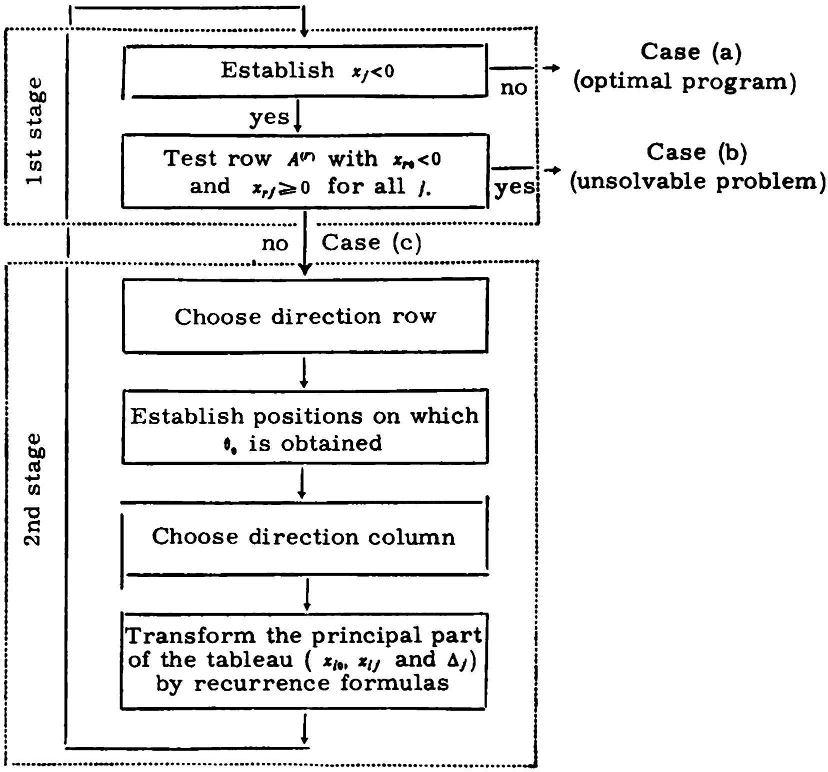


FIGURE 6.5

Now, in some iteration, let the minimum θ_0 in the row θ be obtained simultaneously on several vectors

$$\theta_0 = -\frac{\Delta_{k_1}}{x_{rk_1}} = -\frac{\Delta_{k_2}}{x_{rk_2}} = \ldots = -\frac{\Delta_{k_t}}{x_{rk_t}},$$
$$k_1 < k_2 < \ldots < k_t;$$

where $x_{rk_i} < 0$ $(i=1, 2, \ldots, t)$.

We shall say that $A_{k_1}, \ldots, A_{k_t}$ are s u s p i c i o u s vectors. One of these should be introduced into the basis of the next pseudoprogram.

Let s_{p_1} be the highest of the indices $s_1, \ldots, s_m$ of the current basis vectors. The vector A_{k_1} is introduced into the successive basis if $k_i > s_{p_1}$ for $i=1,$ $2, \ldots, t$. Otherwise, the set of suspicious vectors is narrowed down and in the next test we need consider only those vectors whose indices do not exceed s_{p_1}. For these vectors we construct the row

$$\theta_1 = \left(\frac{x_{p_1 j}}{x_{rj}}\right).$$

The vector on which

$$\theta_{10} = \min \theta_1$$

is obtained (if it is unique) is introduced into the basis. If θ_{10} is also obtained on several vectors, these form a new set of suspicious vectors. Let s_{p_2} be the second highest index of the current basis vectors. The vector with the lowest index is introduced into the successive basis, if no suspicious vectors exist with indices lower than s_{p_2}. Otherwise, the set of suspicious vectors is again narrowed down to comprise only vectors with indices less than s_{p_2}. For these vectors we form the row

$$\theta_2 = \left(\frac{x_{p_2 j}}{x_{rj}} \right),$$

whose minimum elements define the next set of suspicious vectors. The process is continued until the set of suspicious vectors contains a single element. The vector to be introduced into the basis is determined uniquely in at most m steps.

Consider an application of this rule.

TABLE 6.2

No.	B_x	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}
1	A_7	3	−15			7	−1	−4	−1	−8	6			−9
2	A_{10}	−1	5		−1	−4	5	2		−2	1	−1		−2
← 3	A_2	−5	−5	1	−3	−3	2	−2		−1	−1			−1
4	A_{11}	−2	−15			−12	3	−6		4	−3		−1	−5
Δ	−		10	−	9	6	4	4	−	5	2	−	−	2
θ	−	−	**2**	−	3	**2**	—	**2**	−	5	**2**	−	−	**2**
θ_1	−	−	**3**	−	−	4	−	**3**	−	−	**3**	−	−	−
θ_2	−	−	**−1**	−	−	−	−	**−1**	−	−	**−1**	−	−	−
θ_3	−	−	3	−	−	−	−	**2**	−	−	−	−	−	−

In Table 6.2 we give the coefficients in the expansion of the constraint vector and restraint vectors in terms of the basis of some pseudoprogram obtained in one of the iterations in the solution process of a problem by the dual simplex method.

We see that the minimum value of the row $(\theta_0 = 2)$ is obtained simultaneously on five vectors:

$$A_1, \ A_4, \ A_6, \ A_9, \ A_{12}.$$

In this case $r=3$, so that $\theta_j = -\frac{\Delta_j}{x_{3j}}$. Among the basis vectors $A_{s_4} = A_{11}$ is the one with the highest index. Vector A_{12} is, therefore, eliminated from the set of suspicious vectors. The restraint vector A_{11} occupies the fourth position in the basis. We compute the ratio $\frac{x_{4j}}{x_{3j}}$ (elements of the row θ_1) for the remaining suspicious vectors. The least value in the row θ_1 is obtained on the vectors A_1, A_6, and A_9.

The second highest index of basis vectors is $s_2 = 10$. The indices of all the remaining suspicious vectors are smaller than 10. We compute the elements of the row $\theta_2 = \left(\frac{x_{2j}}{x_{3j}}\right)$ for $j=1$, 6, 9. All three θ_2-values are equal. Continuing with this procedure, we eliminate the vector A_9, since its index is higher than the third highest index of the basis vectors, $A_{s_1} = A_7$. We are left with two suspicious vectors A_1 and A_6. Finally, the row θ_3 enables us to establish the index of the vector to be introduced into the basis. This is A_6, for which

$$\theta_{30} = \min\left(\frac{x_{11}}{x_{31}},\ \frac{x_{16}}{x_{36}}\right) = \min(3,\ 2) = 2$$

is obtained.

The dual method can, thus, be equally well applied in degenerate and nondegenerate cases. Degeneracy of the dual problem only complicates the choice of the vector A_k introduced into the basis. However, the number of iterations always remains finite though, generally speaking, the linear form does not decrease monotonically.

It is noteworthy that when solving applied problems by the dual method, as in the case of the simplex method, cycling is most improbable. Moreover, construction of examples illustrating cycling requires a certain ingenuity. Also the method for choosing the vector A_k which guarantees against cycling is comparatively complicated. It is, therefore, advisable in practical computations to use certain simpler rules for choosing the vector to be introduced into the basis. One of these rules states, for example, that the suspicious vector with the lowest (or the highest) index is introduced into the basis. In most cases this elementary rule is sufficient to prevent returning to a basis which has previously been examined. In those rare cases when this simplified rule produces cycling, we should apply the exact rule when the same basis recurs. The exact rule is applied until we obtain a pseudoprogram basis which actually decreases the linear form. We then return to the simplified rule which specifies uniquely the vector to be introduced into the basis.

It should be kept in mind that in each transition from the simplified to the exact rule, the restraint vectors should be renumbered so that the basis in question comprises the vectors A_1, A_2, ..., A_m, and this numeration should be observed at least until the linear form decreases for the first time.

5-3. We shall illustrate the solution of a linear-programming problem by the dual method with two examples.

Example 1. Maximize the linear form

$$L(X) = -(3x_1 + x_2 + 2x_3 + 3x_4 + x_5 + 2x_6 + 5x_7)$$

TABLE 6.3 (0.3)

	No	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	No. of tableau
		C			−3	−1	−2	−3	−1	−2	−5						
←	1		A_8	−10	−2		−3			−2		1					0
	2		A_9	−4		−4			−3	−1			1				
	3		A_{10}	−2		−1	−5		−4		−5			1			
	4		A_{11}	−5	−3			−2		−3					1		
	5		A_{12}	−3			−2	−4			−3					1	
	6	—	Δ		3	1	2	3	1	2	5						
	7	—	θ	—	1.5	—	0.667	—	—	1	—	—	—	—	—	—	
→	1	−2	A_3	−3.333	0.667		1			0.667		−0.333					1
	2		A_9	−4		−4			−3	−1			1				
	3		A_{10}	14.667	3.333	−1			−4	3.333	−5	−1.667		1			
←	4		A_{11}	−5	−3			−2		−3					1		
	5		A_{12}	3.667	1.333			−4		1.333	−3	−0.667				1	
	6	—	Δ	−6.661	1.667	1		3	1	0.667	5	0.667					
	7	—	θ	—	0.556	—	—	1.5	—	0.222	—	—	—	—	—	—	

TABLE 6.3 (continued)

No	C_X	B_X		−3	−1	−2	−3	−1	−2	−5						No. of tableau
No	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	
1	−2	A_3	2.222			1	−0.444				−0.333			0.222		2
← 2		A_9	−2.333	1	−4		0.667	−3				1		−0.333		2
3		A_{10}	9.111		−1		−2.222	−4		−5	−1.667		1	1.111		2
→ 4	−2	A_6	1.667	1			0.667		1					−0.333		2
5		A_{12}	1.444				−4.889			−3	−0.667			0.444	1	2
6	—	Δ	−7.778	1	1		2.556	1		5	0.667			0.222		2
7	—	θ	—	—	0.25	—	—	0.333	—	—	—	—	—	0.667	—	2
1	−2	A_3	2.222			1	−0.444				−0.333			0.222		3
→ 2	−1	A_2	0.583	−0.25	1		−0.167	0.75				−0.25		0.833		3
3		A_{10}	9.694	−0.25			−2.389	−3,25		−5	−1.667	−0.25	1	1.194		3
4	−2	A_6	1.667	1			0.667		1					−0.333		3
5		A_{12}	1.444				−4.889			−3	−0.667			0.444	1	3
6	—	Δ	−8.361	1.25			2.722	0.25		5	0.667			0.139		3
7	—	θ	—	—	—	—	—	—	—	—	—	—	—	—	—	3

subject to the conditions

$$
\begin{aligned}
2x_1 \quad +3x_3 \quad\quad +2x_6 \quad -x_8 \quad\quad\quad\quad &= 10,\\
4x_2 \quad\quad +3x_5+x_6 \quad\quad -x_9 \quad\quad\quad &= 4,\\
x_2 \quad +5x_3 \quad +4x_5 \quad +5x_7 \quad -x_{10} \quad\quad &= 2,\\
3x_1 \quad\quad +2x_4 \quad +3x_6 \quad\quad\quad -x_{11} \quad &= 5,\\
2x_3+4x_4 \quad\quad +3x_7 \quad\quad\quad\quad -x_{12} &= 3,\\
x_j \geqslant 0; \quad j=1,\ 2,\ \ldots,\ 12.&
\end{aligned}
$$

Solution. The dual problem is stated as follows:
Minimize the linear form

$$\tilde{L}\,(Y)=10y_1+4y_2+2y_3+5y_4+3y_5$$

subject to the conditions

$$
\begin{aligned}
2y_1 \quad\quad\quad +3y_4 \quad\quad &\geqslant -3,\\
4y_2 +y_3 \quad\quad\quad\quad &\geqslant -1,\\
3y_1 \quad\quad +5y_3 \quad\quad +2y_5 &\geqslant -2,\\
2y_4+4y_5 &\geqslant -3,\\
3y_2 +4y_3 \quad\quad\quad\quad &\geqslant -1,\\
2y_1 +y_2 \quad\quad +3y_4 \quad\quad &\geqslant -2,\\
5y_3 \quad\quad +3y_5 &\geqslant -5,\\
y_i \leqslant 0; \quad i=1,\ 2,\ 3,\ 4,\ 5.&
\end{aligned}
$$

We see that $Y=(0,\ 0,\ 0,\ 0,\ 0)$ is a support program of the dual problem. The vector Y reduces the last $m=5$ restraint to equalities:

$$y_i=0;$$

the other restraints of the dual problem are reduced by Y to strict inequalities. The basis of the support program Y of the dual problem comprises the restraint vectors

$$A_8=(-1,\ 0,\ 0,\ 0,\ 0),\quad A_9=(0,\ -1,\ 0,\ 0,\ 0),$$
$$A_{10}=(0,\ 0,\ -1,\ 0,\ 0),\ A_{11}=(0,\ 0,\ 0,\ -1,\ 0),\ A_{12}=(0,\ 0,\ 0,\ 0,\ -1).$$

All the computations entailed in the solution of Example 1 are given in Table 6.3 (0–3).

The basis of the initial pseudoprogram comprises unit vectors with reverse sign. The basis components of pseudoprogram X (zeroeth tableau) therefore coincide with the corresponding components of the constraint vector taken with reverse sign, and the elements x_{ij} of columns A_j are equal to the elements of the corresponding columns of the restraint matrix $\|a_{ij}\|$ with reversed signs. In the $(m+1)$-th (sixth) row of the tableau, we write the parameters $\Delta_j=-c_j$ (the basis components of the initial pseudoprogram correspond to zero coefficients of the linear form).

All the basis components of the pseudoprogram are negative and in each row there are some negative x_{ij} $(i=1,2,3,4,5)$. This indicates that we have case (c). We now pass to a new pseudoprogram. The vector $A_{s_1}=A_8$ $(x_8=x_{1,0}=-10<x_{i0},\ i=2,\ 3,\ 4,\ 5)$ is to be eliminated from the basis. In the last row of the zeroeth tableau we write

$$\theta_j=-\frac{\Delta_j}{x_{1j}}$$

for $x_{1j}<0$. The least element θ_0 of the row θ is obtained on A_1. The restraint vector A_1 should be introduced into the pseudoprogram basis in place of the vector A_8. The principal part of the zeroeth tableau is transformed into the principal part of the 1st tableau using recurrence formulas (5.1).

The 1st tableau and computations connected with transformation to the 2nd tableau are analyzed analogously. The pseudoprogram obtained after the third iteration proves to be a feasible program of the problem. The problem is thus solved by the vector

$$X'''=(0,\ 0.583,\ 2.222,\ 0,\ 0,\ 1.667,\ 0,\ 0,\ 0,\ 9.694,\ 0,\ 1.444).$$

The maximum of the linear form is equal to

$$L\,(X''')=-8.361.$$

In Tables 6.4 (0–2) we give the sequence of solution of another example.

Example 2. Maximize the linear form

$$L\,(X)=-(5x_1+3x_2+4x_3+x_4+2x_5+3x_6)$$

TABLE 6.4 (0–2)

C				−5	−3	−4	−1	−2	−3						
No.	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	No. of tableau
1		A_7	−12	−6	−2	−3	−8	−1	−2	1					0
2		A_8	−5	−4	−5	−8	−3	−1	−7		1				
3		A_9	−7	−2			−1	−4	−2			1			
4		A_{10}	−10	−5	−2	−1	−1	−6	−5				1		
5		A_{11}	−4	−1	−3		−6	−3	−2					1	
6	—	Δ		5	3	4	1	2	3						
7	—	θ	—	0.833	1.5	1.333	0.125	2	1.5	—	—	—	—	—	
1	−1	A_4	1.5	0.75	0.25	0.375	1	0.125	0.25	−0.125					1
2		A_8	−0.5	−1.75	−4.25	−6.875		−0.625	−6.25	−0.375	1				
3		A_9	−5.5	−1.25	0.25	0.375		−3.875	−1.75	−0.125		1			
4		A_{10}	−8.5	−4.25	−1.75	−0.625		−5.875	−4.75	−0.125			1		
5		A_{11}	5	3.5	−1.5	2.25		−2.25	−0.5	−0.75				1	
6	—	Δ	−1.5	4.25	2.75	3.625		1.875	2.75	0.125					
7	—	θ	—	1	1.571	5.8	—	0.319	0.579	1	—	—	—	—	
1	−1	A_4	1.319	0.660	0.213	0.362	1		0.149	−0.128			0.021		2
2		A_8	0.404	−1.298	−4.064	−6.809			−5.745	−0.362	1		−0.106		
3		A_9	0.106	1.553	1.404	0.787			1.383	−0.043		1	−0.660		
4	−2	A_5	1.447	0.723	0.298	0.106		1	0.809	0.021			−0.170		
5		A_{11}	8.255	5.128	−0.830	2.489			1.319	−0.702			−0.383	1	
6	—	Δ	−4.213	2.894	2.191	3.426			1.234	0.085			0.319		
7	—	θ	—	—	—	—	—	—	—	—	—	—	—	—	

subject to the conditions

$$\begin{array}{l}
6x_1+2x_2+3x_3+8x_4+x_5+2x_6-x_7 = 12, \\
4x_1+5x_2+8x_3+3x_4+x_5+7x_6 \qquad -x_8 = 5, \\
2x_1 \qquad + x_4+4x_5+2x_6 \qquad -x_9 = 7, \\
5x_1+2x_2+x_3+x_4+6x_5+5x_6 \qquad -x_{10} = 10, \\
x_1+3x_2 \qquad +6x_4+3x_5+2x_6 \qquad -x_{11} = 4, \\
x_j \geqslant 0; \quad j=1, 2, \ldots, 11.
\end{array}$$

It is unnecessary to explain the tableaus.

5-4. We now present the dual simplex method (or, more precisely, its first algorithm), without resorting to the linear-programming problem in the vector form and without introducing the dual problem.

We shall limit the discussion to the nondegenerate case.

We solve the problem restraints (1.2) for the m variables $x_{s_1}, \ldots, x_{s_m}$ corresponding to the basis components $(x_{10}, \ldots, x_{m0})$ of the initial pseudoprogram X. As previously, let I denote the set of indices of the basis vectors.

We have seen in Chapter 5, § 4, that the linear-programming problem (1.1)–(1.3) can be stated in the following equivalent form.

Compute the nonnegative $x_1, x_2, \ldots, x_n$ satisfying the restraints

$$x_{s_i} = x_{i0} - \sum_{j \notin I} x_{ij} x_j, \qquad i=1, 2, \ldots, m, \tag{5.2}$$

and maximizing the linear form

$$L(x) = L(X_0) - \sum_{j \notin I} \Delta_j x_j, \tag{5.3}$$

where X_0 is the vector whose components x_{s_i} are equal to x_{i0} $(i=1, 2, \ldots, m)$, and the other components are all zero. X_0 is a pseudoprogram of problem (1.1)–(1.3). This indicates that the choice of variables $x_{s_1}, \ldots, x_{s_m}$ ensures nonnegativity of all the coefficients of Δ_j in expression (5.3) of linear form $L(X)$.

Let

$$L(X) = x_{s_{m+1}}, \quad L(X_0) = x_{m+1,0}, \quad \Delta_j = x_{m+1,j} \tag{5.4}$$

and consider the system of equations combining equalities (5.2) with (5.3):

$$x_{s_i} = x_{i0} - \sum_{j \notin I} x_{ij} x_j, \qquad i=1, 2, \ldots, m, m+1. \tag{5.5}$$

We solve this system by the Gauss method (complete reduction method). We express x_k $(k \notin I)$ from the r-th equation $(1 \leqslant r \leqslant m)$ and insert the result in the other equations (5.5). It is naturally assumed that $x_{rk} \neq 0$. After the first step of the complete reduction method, we obtain

$$x_{s'_i} = x'_{i0} - \sum_{j \notin I'} x'_{ij} x_j, \qquad i=1, 2, \ldots, m, m+1, \tag{5.6}$$

where $s'_i = s_i$ for $i \neq r$, $s'_r = k$;

$$\left\{ \begin{matrix} I \\ j \neq s_r \end{matrix} \right\} \equiv \left\{ \begin{matrix} I' \\ j \neq k \end{matrix} \right\}.$$

The parameters x'_{ij} are determined from

$$x'_{ij} = \begin{cases} x_{ij} - \dfrac{x_{ik}}{x_{rk}} x_{rj}, & i \neq r, \\ \dfrac{x_{rj}}{x_{rk}}, & i = r, \end{cases} \tag{5.7}$$

$$i=1, 2, \ldots, m, m+1; \quad j=0, 1, 2, \ldots, n.$$

The element x_{rk} is called the d i r e c t i o n e l e m e n t o f t h e t r a n s f o r m a t i o n.

Until now we have imposed no restrictions on the choice of the number r of the equation and the index k of the variable specifying the direction element of the transformation, except for the obvious requirement $x_{rk} \neq 0$. We shall now impose the following restrictions on the choice of r and k.

The index r is specified by the greatest (in absolute value) negative x_{i0},

$$x_{r0} = \min_i x_{i0}$$

(if all $x_{i0} \geqslant 0$, the program is optimal).

Now, there exists at least one feasible program of problem (1.1)–(1.3) if for $x_{r0} < 0$ at least one of the $x_{rj} (j = 1, 2, \ldots, n)$ is negative. Otherwise, as we see from (5.6),

$$x'_{s_r} \leqslant x'_{r0} < 0$$

for any nonnegative extrabasis variables $x_j (j \notin I)$.

The choice of the index k should ensure a transition to a new pseudoprogram X'. This indicates that for the given k

$$\Delta'_j = x'_{m+1,\, j} \geqslant 0, \qquad j = 1, 2, \ldots, n. \tag{5.8}$$

According to recurrence formulas (5.7), condition (5.8) can be rewritten in the form

$$\Delta'_j = \Delta_j - \frac{\Delta_k}{x_{rk}} x_{rj} \geqslant 0, \qquad j = 1, 2, \ldots, n. \tag{5.9}$$

The vector X_0 is by assumption a pseudoprogram. Hence,

$$\Delta_j \geqslant 0, \qquad j = 1, 2, \ldots, n.$$

For $j = s_r$ $x_{rj} = 1$, $\Delta_j = 0$. Hence, from (5.9),

$$-\frac{\Delta_k}{x_{rk}} \geqslant 0.$$

But $\Delta_k > 0$ (a nondegenerate case is being considered). Therefore $x_{rk} < 0$. If now

$$x_{rj} > 0,$$

(5.9) is automatically satisfied.

We have seen that a solvable problem must contain at least one $x_{rj} < 0$. To satisfy condition (5.9) it is necessary that for $x_{rj} < 0$

$$-\frac{\Delta_k}{x_{rk}} \leqslant -\frac{\Delta_j}{x_{rj}},$$

i.e., the index k is specified by the variable x_k for which

$$-\frac{\Delta_k}{x_{rk}} = \min_{x_{rj} < 0} \left(-\frac{\Delta_j}{x_{rj}} \right). \tag{5.10}$$

We now compare systems (5.5) and (5.6) with the tableaus of two successive iterations of the first dual algorithm. We see that the elements of the right-hand sides of the equations are arranged in the same order and are transformed according to the same recurrence formulas as the corresponding elements in the tableaus.

The dual simplex method, with a given initial pseudoprogram, thus reduces to solution of systems of equations by the Gauss complete reduction method with special rules imposed on the choice of the direction element x_{rk}. The choice of r is restricted by

$$x_{r0} = \min_i x_{i0},$$

and the k chosen should satisfy (5.10). In other words, the choice of the direction row ensures monotonic variation of the linear form, and the choice of the direction column, motion over pseudoprograms.

5-5. Let us now consider the specific features appearing when the first dual algorithm is applied to problems with bilateral restraints.

The modifications introduced in the computational procedure because of the bilateral restraints apply mainly to the method of determining the direction element of the transformation and to the formulas for computation of the basis components of the pseudoprogram. This also causes some modifications in the structure of the first-algorithm tableaus.

The l-th tableau corresponding to the l-th iteration in a problem with bilaterally restrained variables differs from the l-th tableau of the problem in canonical form in the additional row X and in three additional columns $\tilde{A}_0$, $(\alpha; \beta)$, and δ (Table 6.5).

In row X, under each of the restraint vectors not appearing in the basis, we write α, if $x_j^{(l)}$ assumes the left boundary value (i.e., if $\Delta_j>0$), and β if $x_j^{(l)}$ is equal to the right boundary value (i.e., if $\Delta_j<0$). The cells of row X corresponding to the basis vectors are crossed out.

To the right of the tableau columns corresponding to the restraint vectors we add two columns, $(\alpha; \beta)_X$ and δ. In column $(\alpha; \beta)_X$ we write the boundary values for each of the basis variables. The entries in column δ are equal to the deviations $\delta_i^{(l)}$ of the basis components of the pseudoprogram from the boundary values of the corresponding variables:

$$\delta_i^{(l)} = \begin{cases} \alpha_{s_i} - x_{i0}^{(l)}, & \text{if } x_{i0}^{(l)} < \alpha_{s_i}, \\ x_{i0}^{(l)} - \beta_{s_i}, & \text{if } x_{i0}^{(l)} > \beta_{s_i}. \end{cases} \tag{5.11}$$

The cells of column δ for which

$$\alpha_{s_i} \leqslant x_{i0}^{(l)} \leqslant \beta_{s_i},$$

are crossed out. The numerical values of $\delta_i^{(l)}$ should be modified by α or β depending on whether x_{i0} is on the left or the right of the permissible interval of the corresponding variable.

The vector A_{s_r} corresponding to the maximum deviation $\delta_r^{(l)}$ should be eliminated from the basis.

The modification α or β added to $\delta_r^{(l)}$ indicates which of the cases, (i) or (ii), applies for the given iteration (see 3-5).

In (i), the elements of the last row θ are computed from

$$\theta_j^{(l)} = -\frac{\Delta_j^{(l)}}{x_{rj}^{(l)}}. \tag{5.12}$$

Actual entries are made only in those cells of the row θ for which

$$\frac{\Delta_j^{(l)}}{x_{rj}^{(l)}} < 0, \quad x_{rj} \neq 0.$$

The remaining cells of the row θ, as those corresponding to the basis vectors, are crossed out. In the degenerate case, when $\Delta_j = 0$ from some A_j not appearing in the basis, we must treat Δ_j as a positive number for $x_j = \alpha_j$ and as a negative for $x_j = \beta_j$.

In (ii) the entries in row θ are computed from

$$\theta_j^{(l)} = \frac{\Delta_j^{(l)}}{x_{rj}^{(l)}}. \tag{5.13}$$

TABLE 6.5

l-th tableau

No.	C_X	B_X	A_0	$\tilde{A}_0$	A_1	A_2	...	A_k	...	A_n	δ	$(\alpha; \beta)_X$
1	c_{s_1}	A_{s_1}	$x_{10}^{(l)}$	$\tilde{x}_{10}^{(l)}$	$x_{11}^{(l)}$	$x_{12}^{(l)}$	...	$x_{1k}^{(l)}$	...	$x_{1n}^{(l)}$		$\alpha_{s_1}; \beta_{s_1}$
2	c_{s_2}	A_{s_2}	$x_{20}^{(l)}$	$\tilde{x}_{20}^{(l)}$	$x_{21}^{(l)}$	$x_{22}^{(l)}$	...	$x_{2k}^{(l)}$	...	$x_{2n}^{(l)}$		$\alpha_{s_2}; \beta_{s_2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	...	$\vdots$	...	$\vdots$
r	c_{s_r}	A_{s_r}	$x_{r0}^{(l)}$	$\tilde{x}_{r0}^{(l)}$	$x_{r1}^{(l)}$	$x_{r2}^{(l)}$	...	$x_{rk}^{(l)}$	...	$x_{rn}^{(l)}$	$\delta_r^{(l)}\ (\gamma_r)$	$\alpha_{s_r}; \beta_{s_r}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	...	$\vdots$	$\vdots$	$\vdots$
m	c_{s_m}	A_{s_m}	$x_{m0}^{(l)}$	$\tilde{x}_{m0}^{(l)}$	$x_{m1}^{(l)}$	$x_{m2}^{(l)}$	...	$x_{mk}^{(l)}$	...	$x_{mn}^{(l)}$		$\alpha_{s_m}; \beta_{s_m}$
$m+1$	—	Δ	$L^{(l)}$	—	$\Delta_1^{(l)}$	$\Delta_2^{(l)}$	...	$\Delta_k^{(l)}$	...	$\Delta_m^{(l)}$	—	—
$m+2$	—	X	—	—			...	γ_k	...		—	—
$m+3$	—	θ	—	—			...	θ_0	...		—	—

Actual entries are made only in those cells where

$$\frac{\Delta_j^{(l)}}{x_{rj}^{(l)}}>0, \quad x_{rj}^{(l)} \neq 0.$$

The other cells of the row θ are crossed out. As in (i), the zero Δ_j arising for extrabasis variables in the analysis of degenerate programs are treated as positive for $x_j=\alpha_j$ and as negative for $x_j=\beta_j$.

The column A_0 contains the basis components of the pseudoprogram (the coefficients in the expansion of the vector

$$A_{\bullet}=B-\sum_{j \notin I_Y} x_j A_j$$

in terms of the basis vectors of the pseudoprogram). The vector $A_{\bullet}$ changes from one basis to another and, therefore, the general recurrence formulas used in transforming the tableau do not apply to it. However, it can be verified that the basis components of the pseudoprogram in the $(l+1)$-th iteration are related by recurrence formulas (5.1) with the components of the vector

$$\tilde{A}_{\bullet}=A_{\bullet}+x_k^{(l)}A_k-\gamma_r^{(l)}A_{s_r}, \tag{5.14}$$

corresponding to the l-th iteration. Here s_r is the index of the vector A_{s_r} eliminated from the basis in the l-th iteration, k is the index of the vector introduced into the basis in the l-th iteration; $\gamma_r^{(l)}$ is equal to α_{s_r} or β_{s_r} depending on the modification (α or β) of the deviation $\delta_r^{(l)}$.

The column $\tilde{A}_{\bullet}$ is inserted into the tableau between columns $A_{\bullet}$ and A_1. The elements of column $\tilde{A}_{\bullet}$ are computed from

$$\tilde{x}_{i0}^{(l)}=\begin{cases} x_{i0}^{(l)}+x_k^{(l)}x_{ik}^{(l)} & \text{for } i\neq r, \\ x_{r0}^{(l)}+x_k^{(l)}x_{rk}^{(l)}-\gamma_r^{(l)} & \text{for } i=r. \end{cases} \tag{5.15}$$

The columns $\tilde{A}_0, A_1, \ldots, A_n$ (all $(m+1)$ positions) constitute the principal part of the tableau.

The initial zeroeth tableau differs from all the subsequent tableaus in having two additional rows. In row C we write the linear form coefficients of the problem, and in the row $(\alpha;\beta)$ the boundary values of the variables x_j.

Let us now briefly outline the sequence of computations in a single $(l+1)$-th iteration.

At the beginning of the $(l+1)$-th iteration the l-th tableau is assumed to be filled (except for the column $\tilde{A}_{\bullet}$ and the row θ). The iteration starts by testing pseudoprogram $X^{(l)}$ for optimality. Pseudoprogram $X^{(l)}$ is a feasible program and thus a solution of the problem (case (a)) if all the positions in column δ of the l-th tableau are crossed out. If (case (a)) does not apply, we should check for solvability of the problem. The problem is unsolvable (case (b)) if there is a row with a deviation $\delta_i^{(l)}(\alpha)$ satisfying condition (3.19) or a row with a deviation $\delta_i^{(l)}(\beta)$ satisfying condition (3.20). In other words, the problem is unsolvable if in a row with a deviation modified by α the elements $x_{ij}\geqslant 0$ for $x_j=\alpha_j$ and $x_{ij}\leqslant 0$ for $x_j=\beta_j$ ($j\notin I_Y$) or in a row with a deviation modified by β the elements $x_{ij}\leqslant 0$ for $x_j=\alpha_j$ and $x_{ij}\geqslant 0$ for $x_j=\beta_j$ ($j\notin I_Y$). Here x_j are the elements specified by the row X of the l-th tableau.

If the unsolvability test is negative (case (c)), we determine the direction element of the transformation and construct the $(l+1)$-th tableau (except for column $\tilde{A}_0$ and row θ), containing the initial data for the next iteration.

The vector A_{s_r} with the maximum deviation $\delta_r^{(l)}$ is eliminated from the basis. If there are several such vectors, any of them can be eliminated. To determine the vector to be introduced into the basis, we fill the row according to the preceding rules. The least element θ_0 of the row θ specifies the vector A_k to be introduced into the basis. If θ_0 is simultaneously obtained on several elements, the vector with the lowest index is introduced into the basis. The exact rule for choosing the vector A_k which guarantees against cycling is given in 4-4.

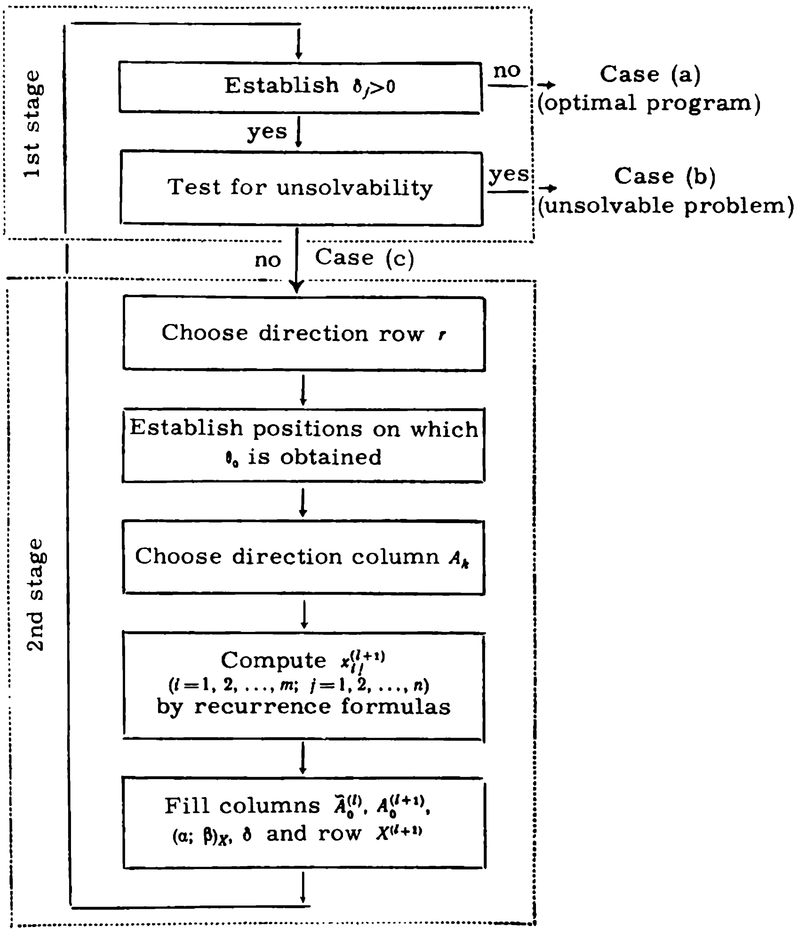


FIGURE 6.6

Given the index of the direction row r and the index of the direction column k, we may compute the elements of the column $\tilde{A}_0$ from (5.14) or (5.15). We now have all the data for passing to the $(l+1)$-th tableau.

The columns A_1, A_2, ..., A_n (positions 1, 2, ..., m, $m+1$) are transformed

according to the recurrence formulas

$$x_{ij}^{(l)}=\begin{cases} x_{ij}^{(l)}-\dfrac{x_{rj}^{(l)}}{x_{rk}^{(l)}}\,x_{ik}^{(l)}, & i\neq r,\\[2mm] \dfrac{x_{rj}^{(l)}}{x_{rk}^{(l)}}, & i=r,\end{cases} \tag{6.16}$$
$$j=1,\ 2,\ \ldots,\ n.$$

The elements of column A_0 of the $(l+1)$-th tableau are computed with the aid of the same recurrence formulas from the entries of column $\tilde{A}_0$ in the l-th tableau:

$$x_{i0}^{(l+1)}=\begin{cases} \tilde{x}_{i0}^{(l)}-\dfrac{\tilde{x}_{r0}^{(l)}}{x_{rk}^{(l)}}\,x_{ik}, & i\neq r,\\[2mm] \dfrac{\tilde{x}_{r0}^{(l)}}{x_{rk}^{(l)}}, & i\neq r.\end{cases} \tag{5.17}$$

The value of the linear form is transformed between two successive iterations according to the formula

$$L\left(X^{(l+1)}\right)=L\left(X^{(l)}\right)-\theta_0^{(l)}\delta_r^{(l)}. \tag{5.18}$$

The entries in row X in the $(l+1)$-th tableau coincide with the corresponding entries of row X in the l-th tableau for all j, except for $j=k$ and $j=s_r$. The $j=k$ position of the row X is crossed out (the vector A_k appears in the basis of pseudoprogram $X^{(l+1)}$). In the s_r-th position the modification (α or β) of $\delta_r^{(l)}$ from the l-th tableau is written.

In column $(\alpha;\ \beta)_X$ of the $(l+1)$-th tableau we write the same figures as in column $(\alpha;\ \beta)_X$ of the l-th tableau. The only exception is the r-th position, where $(\alpha_{s_r};\ \beta_{s_r})$ is substituted for $(\alpha_k;\ \beta_k)$. The column δ is filled according to formulas (5.11) following the preceding remarks.

Determination of all these parameters completes the $(l+1)$-th iteration. Subsequent iterations follow the same rules.

The computations should be checked by determining Δ_j and L from recurrence formulas and directly from the relationships defining these parameters.

Figure 6.6 is a block diagram of the solution of a linear-programming problem with bilateral restraints by the first dual algorithm.

5-6. We will now illustrate the application of this computational procedure by an appropsiate example. Determine the vector X maximizing the linear form

$$L=12x_1+2x_2+10x_3+14x_4+x_6+2x_7$$

subject to the conditions

$$\begin{aligned} 2x_1+\ x_2+3x_3+5x_4+x_5\qquad\qquad &=1,\\ x_1+4x_2+2x_3+3x_4\quad +x_6\qquad &=4,\\ 4x_1+2x_2+3x_3+5x_4\qquad +x_7\quad &=4,\\ 3x_1+2x_2+4x_3+\ x_4\qquad\qquad +x_8&=8,\\ -1\leqslant x_j\leqslant 1,\quad j=1,\ 2,\ \ldots,\ 8.& \end{aligned}$$

The process of solution of the problem is given in Table 6.6 (0-2). All the variables are bilaterally restrained. Therefore, any system of $m=4$ linearly independent vectors can be taken as the basis of the initial pseudoprogram. It is natural to choose the unit vectors A_5, A_6, A_7, and A_8 as the initial basis. In the columns $A_1, A_2, \ldots, A_8$ (the first $m=4$ positions) of the zeroeth tableau the components of the restraint vectors are written. The elements of the row Δ are computed from (1.11). In row X we write the extrabasis variables. For $\Delta_j>0$, $x_j=\alpha_j$, for $\Delta_j<0$, $x_j=\beta_j$ Here

$$x_2=\alpha_2=-1,\quad x_1=\beta_1=x_3=\beta_3=x_4=\beta_4=1.$$

TABLE 6.6 (0–2)

					C	12	2	10	14		1	2				
					$\alpha;\ \beta$	−1; 1	−1; 1	−1; 1	−1; 1	−1; 1	−1; 1	−1; 1	−1; 1			No. of tableau
	No.	C_X	B_X	A_0	$\tilde{A}_0$	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	δ	$(\alpha;\ \beta)_X$	
←	1		A_5	−8	−2	2	1	3	5	1				7(α)	−1; 1	0
	2	1	A_6	2	5	1	4	2	3		1			1(β)	−1; 1	
	3	2	A_7	−6	−1	4	2	3	5			1	•	5(α)	−1; 1	
	4		A_8	2	3	3	2	4	1				1	1(β)	−1; 1	
	5	—	Δ	24	—	−3	6	−2	−1					—	—	
	6	—	X	—	—	β	α	β	β	—	—	—	—	—	—	
	7	—	θ	—	—	1.5	—	0.667	0.2	—	—	—	—	—	—	

TABLE 6.6 (continued)

					C	12	2	10	14		1	2				
					α; β	−1; 1	−1; 1	−1; 1	−1; 1	−1; 1	−1; 1	−1; 1	−1; 1			No. of tableau
	No.	C_X	B_X	A_0	$\tilde{A}_0$	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	δ	$(\alpha;\beta)_X$	
→	1	14	A_4	−0.4	−0.6	0.4	0.2	0.6	1	0.2				—	−1; 1	1
←	2	1	A_6	6.2	1.8	−0.2	3.4	0.2		−0.6	1			5.2(β)	−1; 1	
	3	2	A_7	1		2	1			−1		1		—	−1; 1	
	4		A_8	3.4	1.6	2.6	1,8	3,4		−0.2			1	2.4(β)	−1; 1	
	5	—	Δ	22.6	—	−2.6	6.2	−1,4		0.2				—	—	
	6	—	X	—	—	β	α	β	—	α	—	—	—	—	—	
	7	—	θ	—	—	13	1.824	—	—	—	—	—	—	—	—	
	1	14	A_4	−0.706		0.412		0,588	1	0.235	−0.059				−1; 1	2
→	2	2	A_2	0.529		−0.059	1	0.059		−0.176	0.294				−1; 1	
	3	2	A_7	−0.529		2.099		−0.059		−0.824	−0,294	1			−1; 1	
	4		A_8	0.647		2.706		3.294		0.118	−0,529		1		−1; 1	
	5	—	Δ	13.118	—	−2.235		−1.765		1.294	−1.824			—	—	
	6	—	X	—	—	β	—	β	—	α	β	—	—	—	—	
	7	—	θ	—	—		—		—			—	—	—	—	

In column A_0 we write the basis variables of the pseudoprogram

$$A_0 = B - \sum_{j \notin I_Y} x_j A_j.$$

We have, for instance,

$$x_{20} = b_2 - (\beta_1 a_{21} + \alpha_2 a_{22} + \beta_3 a_{23} + \beta_4 a_{24}) = 4 - (1 \cdot 1 - 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 3) = 2.$$

The value of the linear form on the pseudoprogram is equal, by definition, to

$$L(X) = \sum_{j=1}^{8} c_j x_j = 12 \cdot 1 + 2 \cdot (-1) + 10 \cdot 1 + 14 \cdot 1 + 0 \cdot (-8) + 12 + 2 \cdot (-6) + 0 \cdot 2 = 24.$$

The elements of column δ are computed from (5.11). Column δ contains some entries. Hence, the initial pseudoprogram does not solve the problem. The unsolvability test is also negative. We have thus case (c).

The vector $A_{s_1} = A_5$ corresponds to the largest deviation $\delta_1 = 7$ and so should be eliminated from the basis. δ_1 is modified by α. The row θ is, therefore, computed from (5.12). Only the three positions corresponding to the vectors A_1, A_3, and A_4, for which

$$\frac{\Delta_j}{x_{rj}} = \frac{\Delta_j}{x_{1j}} < 0$$

are filled in the row. The least value θ_0 in the row θ is 0.2, which is obtained on the vector A_4. The restraint vector A_4 should thus be introduced into the successive basis. The direction row of the transformation is thus the first row $(r=1)$, and the direction column is column A_4 $(k=4)$. We are now in a position to compute the entries of column $\bar{A}_0$ from (5.15) and to pass to the 1st tableau. The elements x'_{ij} $(i=1, \ldots, 5;$ $j=1, \ldots, 8)$ are computed from recurrence formulas (5.16). The basis components of the successive pseudoprogram are determined from (5.17) using the elements of column $\bar{A}_0$. The linear form on pseudoprogram X' is equal, according to (5.18), to

$$L(X) = 24 - 0.2 \cdot 7 = 22.6.$$

In row X the cell corresponding to A_4 is crossed out, and in the position corresponding to A_5 we write the modifier α accompanying the deviation δ_1 in the zeroeth tableau. In all the other positions in row X the same modifiers as in the zeroeth tableau appear.

Analysis of pseudoprogram X' and determination of the parameters of the successive pseudoprogram follow the same rules. Observe that in the first iteration we have (i), and in the second case, (ii).

In the 2nd tableau all the cells in column δ are crossed out. Pseudoprogram X'' thus solves the problem:

$$X'' = (1;\ 0.529;\ 1;\ -0.706;\ -1;\ 1;\ -0.529;\ 0.647);$$
$$L(X'') = 13.118.$$

§ 6. The second dual algorithm

6-1. The second dual algorithm differs from the first algorithm in the same aspects in which the second simplex algorithm differs from the corresponding first algorithm.

Let us first apply the second algorithm to the canonical form of a problem.

The solution of the problem starts with some given initial support program of the dual problem. The computations are arranged exactly as in the second simplex algorithms, i.e., in a series of principal tableaus and one auxiliary tableau.

In the principal tableaus (Table 6.7) we write the basis components $e_{i0}^{(l)}$ of the pseudoprogram and the coefficients $e_{ij}^{(l)}$ in the expansion of the m-dimensional unit vectors

$$e_j = \underbrace{(\overbrace{0, \ldots, 0, 1}^{j}, 0, \ldots, 0)}_{m}$$

in terms of the vectors of the dual basis. The $(m+1)$-th row of the principal

tableau contains the value of the linear form

$$e_{m+1,0}^{(l)} = \sum_{i=1}^{m} b_i y_i^{(l)}$$

and the components

$$e_{m+1,i}^{(l)} = y_i^{(l)}$$

of the support program of the dual problem. If only the initial dual basis is given, y_j are computed from the formula

$$y_j = e_{m+1,j} = \sum_{i=1}^{m} c_{s_i} e_{ij}, \qquad j = 1, 2, \ldots, m. \tag{6.1}$$

The columns $e_0, e_1, \ldots, e_m$ (all $(m+1)$ positions) constitute the principal part of the tableau. When passing from a given tableau to a successive one, the principal part of the tableau is transformed according to the same recurrence formulas (5.5)-(5.6), Chapter 5, as the elements of the principal tableaus in the second simplex algorithm.

TABLE 6.7

The principal l-th tableau

No.	C_X	B_X	e_0	e_1	e_2	$\cdots$	e_m	$A_k^{(l)}$
1	c_{s_1}	A_{s_1}	$e_{10}^{(l)}$	$e_{11}^{(l)}$	$e_{12}^{(l)}$	$\cdots$	$e_{1m}^{(l)}$	$x_{1k}^{(l)}$
2	c_{s_2}	A_{s_2}	$e_{20}^{(l)}$	$e_{21}^{(l)}$	$e_{22}^{(l)}$	$\cdots$	$e_{2m}^{(l)}$	$x_{2k}^{(l)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
r	c_{s_r}	A_{s_r}	$e_{r0}^{(l)}$	$e_{r1}^{(l)}$	$e_{r2}^{(l)}$	$\cdots$	$e_{rm}^{(l)}$	$x_{rk}^{(l)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
m	c_{s_m}	A_{s_m}	$e_{m0}^{(l)}$	$e_{m1}^{(l)}$	$e_{m2}^{(l)}$	$\cdots$	$e_{mm}^{(l)}$	$x_{mk}^{(l)}$
$m+1$	—	—	$L^{(l)}$	$y_1^{(l)}$	$y_2^{(l)}$	$\cdots$	$y_m^{(l)}$	$\Delta_k^{(l)}$

Observe that in the second simplex algorithm the $(m+1)$-th row is required for subsequent computation of the parameters $\Delta_j^{(l)}$ in each iteration. In the second dual algorithm, the parameters $\Delta_j^{(l)}$, as will be shown in the following, are computed from recurrence formulas. The elements $e_{m+1,i} = y_i$ are, therefore, required only in the zeroeth tableau for computing the initial Δ_j. In all the other principal tableaus, the $(m+1)$-th row is of secondary importance, as it is used for computational checks.

The last column $(A_k^{(l)})$ of the principal tableau contains the coefficients $x_{ik}^{(l)}$ in the expansion of the vector A_k to be introduced into the basis in terms of the basis vectors of the pseudoprogram. The components $x_{ik}^{(l)}$ are computed from

$$x_{ik}^{(l)} = \sum_{s=1}^{m} e_{is}^{(l)} a_{sk}. \tag{6.2}$$

The entries of column A_k are required for transforming the principal part

TABLE 6.8

Auxialiary tableau

No.	B	A_1	A_2	...	A_k	...	A_n
1	b_1	a_{11}	a_{12}	...	a_{1k}	...	a_{1n}
2	b_2	a_{21}	a_{22}	...	a_{2k}	...	a_{2n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	...	$\vdots$
m	b_m	a_{m1}	a_{m2}	...	a_{mk}	..	a_{mn}
$m+1$	c_j	c_1	c_2	...	c_k	...	c_n
0	Δ	Δ_1	Δ_2	...	Δ_k	...	Δ_n
	X_r	x_{r1}	x_{r2}	...	x_{rk}	...	x_{rn}
	θ			...	θ_0	...	
1	Δ'	Δ_1'	Δ_2'	...	Δ_k'	...	Δ_n'
	X_r'	x_{r1}'	x_{r2}'	...	x_{rk}'	...	x_{rn}'
	θ'			...		...	
2	Δ''	Δ_1''	Δ_2''	...	Δ_k''	...	Δ_n''
	X_r''	x_{r1}''	x_{r2}''	...	x_{rk}''	...	x_{rn}''
	θ''			...		...	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	...	$\vdots$

2154

of the tableau according to the recurrence formulas

$$e_{ij}^{(l+1)} = \begin{cases} e_{ij}^{(l)} - \dfrac{x_{ik}^{(l)}}{x_{rk}^{(l)}} e_{rj}^{(l)}, & i \neq r, \\ \dfrac{e_{rj}^{(l)}}{x_{rk}^{(l)}}, & i = r, \end{cases} \tag{6.3}$$

$$i = 1, 2, \ldots, m, m+1; \; j = 0, 1, 2, \ldots, m; \; x_{m+1,k}^{(l)} = \Delta_k^{(l)}.$$

The upper part of the auxiliary tableau contains the restraint vectors. In the $(m+1)$-th row we write the linear-form coefficients.

To each principal tableau correspond three rows in the lower part of the auxiliary tableau (Table 6.8): the row $X_r^{(l)}$, the row $\Delta_j^{(l)}$, and the row $\theta^{(l)}$.

The entries of the row $\Delta_j^{(l)}$ are computed from

$$\Delta_j^{(l)} = \sum_{i=1}^{m} a_{ij} y_i^{(l)} - c_j = \sum_{i=1}^{m} a_{ij} e_{m+1,i}^{(l)} - c_j \tag{6.4}$$

for $l = 0$. The parameters $\Delta_j^{(l)}$ of subsequent iterations are determined recurrently.

In the row $X_r^{(l)}$ we write the coefficient $x_{rj}^{(l)}$ in the expansion of the vectors A_j in terms of the dual basis, corresponding to the vector A_{s_r}. The coefficients $x_{rj}^{(l)}$ are computed from

$$x_{rj}^{(l)} = \sum_{i=1}^{m} e_{ri}^{(l)} a_{ij}, \qquad j = 1, 2, \ldots, n. \tag{6.5}$$

Here r is the basis position from which a vector is eliminated in the $(l+1)$-th iteration.

In the row $\theta^{(l)}$ entries are made only in those positions for which $x_{rj}^{(l)} < 0$. Each entry of the row $\theta^{(l)}$ is equal to minus the ratio of the entries in the two preceding rows. The least entry in the row $\theta^{(l)}$, equal to $-\dfrac{\Delta_k^{(l)}}{x_{rk}^{(l)}}$, is denoted by $\theta_0^{(l)}$. The vector A_k, on which $\theta_0^{(l)}$ is obtained, is introduced into the successive basis.

If θ_0 is obtained simultaneously on several vectors A_{k_i}, any of these can be introduced into the basis, e.g., the vector with the lowest index. However, this rule for choosing the vector to be introduced into the basis does not guarantee against cycling. When a cycle is detected, i.e., when a basis which has previously been examined comes up again for consideration, we should choose the next vector to be introduced into the basis according to the exact rule discussed in § 4. The application of this rule to the second algorithm does not differ from its application to the first dual algorithm. However, in the first algorithm, the coefficients x_{ij} (see 5-2 where the application of the rule is illustrated) are contained in the tableau proper, whereas in the second algorithm the required parameters x_{ij} have to be computed from formulas (6.5). This fact complicates the application of the rule to guarantee against cycling when solving problems with degenerate programs by the second dual algorithm.

Computation of the row $X_r^{(l)}$ is the most complicated operation under the second dual algorithm. The determination of each entry $x_{rj}^{(l)}$ requires m multiplications. However, as we have already indicated, the introduction of row $X_r^{(l)}$ eliminates the necessity of computing the parameters Δ_j from cumbersome formulas (6.4). Indeed, according to recurrence formula (5.1),

we have for $i=m+1$ $(x_{m+1,\,j}=\Delta_j)$

$$\Delta_j^{(l+1)}=\Delta_j^{(l)}-\frac{x_{rj}^{(l)}}{x_{rk}^{(l)}}\,\Delta_k^{(l)},\qquad j=0,\,1,\,2,\,\ldots,\,n, \tag{6.6}$$

or, in our notations,

$$\Delta_j^{(l+1)}=\Delta_j^{(l)}+\theta_0^{(l)}x_{rj}^{(l)},\qquad j=0,\,1,\,2,\,\ldots,\,n, \tag{6.7}$$

i. e., the row $\Delta^{(l+1)}$ is obtained as the sum of row $\Delta^{(l)}$ and row $X_r^{(l)}$ multiplied by the chosen element $\theta_0^{(l)}$ of row $\theta^{(l)}$.

6-2. Let us now briefly outline the sequence of operations under the second simplex algorithm.

The computational procedure of the second algorithm involves successive iterations carried out according to the same rules. Each iteration consists of two stages. In the first stage, the pseudoprogram is tested for optimality (case (a)). If the pseudoprogram is not a feasible program of the problem, test for unsolvability (case (b)). This completes the first stage of the iteration. If case (c) obtains, we proceed with the second stage, where we compute the successive pseudoprogram and all its characteristic parameters.

Consider the sequence of computations in the $(l+1)$-th iteration. Testing pseudoprogram $X^{(l)}$ for optimality, we examine the signs of the entries in column e_0 of the principal l-th tableau. If all the entries $e_{i0}^{(l)}$ $(i=1,\,2,\,\ldots,\,m)$ are nonnegative, the pseudoprogram is a feasible program and simultaneously a solution of the problem. Now let some of the $e_{i0}^{(l)}$ be negative. The vector A_{s_r} with the greatest (in absolute value) negative component $e_{r0}^{(l)}$ is chosen as the vector to be eliminated from the basis.

Next we compute the entries of the rows $\Delta^{(l)}$ and $X_r^{(l)}$ of the auxiliary tableau. The row $\Delta^{(l)}$ is filled according to recurrence formulas (6.7) from the parameters of the three preceding rows $(\Delta^{(l-1)},\ X_r^{(l-1)}$ and $\theta^{(l-1)})$.

We should emphasize that the indices r in $X_r^{(l-1)}$ and $X_r^{(l)}$ correspond to different positions of the dual basis. In the former case, r is the basis position from which a vector is eliminated in the l-th iteration, and in the latter case, the index r indicates the basis position from which a vector is eliminated in the $(l+1)$-th iteration. The exact notations $X_{r_l}^{(l)}$ and $X_{r_{l-1}}^{(l-1)}$ would have unnecessarily complicated the formulas.

The entries $x_{rj}^{(l)}$ of the row $X_r^{(l)}$ are computed from (6.5) as the products of the r-th row of the principal l-th tableau by the corresponding columns A_j from the upper part of the auxiliary tableau. The row $X_r^{(l)}$ enables us to test the problem for unsolvability. The linear-programming problem (1.1)-(1.2) has no feasible programs, if all the entries in row $X_r^{(l)}$ are nonnegative (case (b)).

Note that under the second algorithm unsolvability of the problem, if it is indeed unsolvable, is established after more iterations than under the first algorithm. This is due to the fact that under the second algorithm we compute not all the coefficients x_{ij}, as in the first algorithm, but only those x_{rj}, where r is the index of the greatest (in absolute value) negative component of the pseudoprogram.

Case (c) obtains if the row $X_r^{(l)}$ contains at least one negative element. In this case, we proceed with the second stage of the iteration, i. e., choose the vector A_k to be introduced into the basis, expand it in terms of the basis of pseudoprogram $X^{(l)}$ and compute the main part of the $(l+1)$-th principal tableau.

To choose the vector A_k, we compute the row $\theta^{(l)}$ of the auxiliary tableau. As we have already indicated, the entries

$$\theta_j^{(l)} = -\frac{\Delta_j^{(l)}}{x_{rj}^{(l)}}$$

are determined only for $x_{rj}^{(l)} < 0$. The vector A_k on which $\theta_0^{(l)}$, the element with the least value in the row $\theta^{(l)}$, is obtained is then introduced into the basis. To ensure unique choice of A_k in the degenerate case we should follow the simplified or the exact rule, according to the considerations in 6-1.

In the last column $A_k^{(l)}$ of the principal l-th tableau we write the coefficients $x_{ik}^{(l)}$ in the expansion of the vector A_k in terms of the basis vectors of pseudoprogram $X^{(l)}$. The parameters $x_{ik}^{(l)}$ are computed from formulas (6.2) as products of the i-th row from the main part of the l-th principal tableau and the k-th column from the upper part of the auxiliary tableau.

We now have all the necessary data for transforming, with the aid of formulas (6.3), the main part of the l-th principal tableau into the main part of the $(l+1)$-th tableau. In other words, we now have all the initial parameters for the $(l+2)$-th iteration.

The process of constructing the principal tableaus and extending the auxiliary tableau produces after a finite number of iterations the optimal programs of the dual pair of problems or else establishes their unsolvability.

Computations under the second dual algorithms can be carried out in two possible ways. The first of these is based on the simultaneous computation of the parameters

$$e_{m+1,\,j} = y_j \quad (j = 0, 1, \ldots, m)$$

from recurrence formulas (6.3) and directly from (6.1).

Checking in the above case is carried out by computing Δ_j from (6.4) and from (6.7).

In Figure 6.7 we show a block diagram of the solution of a linear-programming problem by the second dual algorithm.

6-3. The computations of a single iteration of the second dual algorithm are slightly more cumbersome than those of a single-iteration of the second simplex algorithm. When compiling the principal tableaus in the dual method, the column θ is not computed. Moreover, we have m multiplications less by not computing the $(m+1)$-th row in the principal tableau. On the other hand, in each iteration of the dual method the auxiliary tableau is extended by three rows, instead of one in the second simplex algorithm. Computation of the row $\Delta^{(l)}$ from recurrence formulas requires $n-m$ multiplications (the Δ_j for the basis vectors are all zero). Compilation of the row $X_r^{(l)}$ entails $(n-m)m$ multiplications (the $(x_{rj}^{(l)}$ for the basis vectors are not computed; $x_{ri_r}^{(l)} = 1$, and $x_{ri_t}^{(l)} = 0$ for $t = 1, 2, \ldots, m$; $t \neq r$). The same number of multiplications was needed in the second simplex algorithm for computing Δ_j. Finally, the row $\theta^{(l)}$ is computed with less than $(n-m)$ divisions.

Note that in the simplex method there is certain freedom in the choice of the vector A_k introduced into the basis. The vector A_k can, generally speaking, be taken as any vector A_j with a negative evaluation Δ_j. Therefore, in the second simplex algorithm we may economize on multiplications by computing the parameters $\Delta_j^{(l)}$ until the first negative evaluation appears, $\Delta_k^{(l)} < 0$. In the dual method (the nondegenerate case), the vector A_k is

uniquely specified by the least element $\theta_0^{(l)}$ in the row $\theta^{(l)}$. To form the row $\theta^{(l)}$ we must compute all the elements of rows $\Delta_j^{(l)}$ and $X_r^{(l)}$. From the above we see that in a single iteration the second simplex algorithm has some advantage.

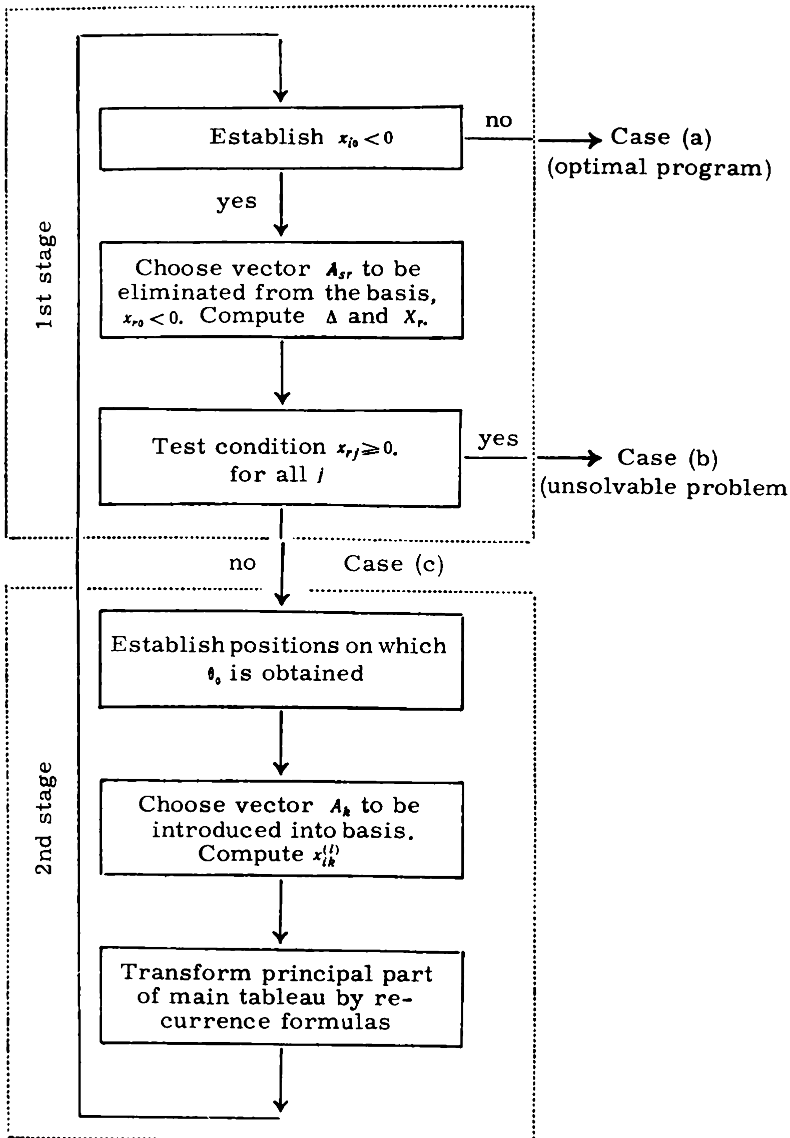


FIGURE 6.7

As we have already indicated in our discussion of the first algorithm, no evaluations are available for comparing the total number of iterations required for obtaining a solution by the two methods.

Comparing the first and second dual algorithms we reach the same conclusions as when comparing the two simplex algorithms.

6-4. Let us now illustrate the solution of a linear-programming problem by the second dual algorithm with the example considered in 5-3 in connection with the first algorithm.

The entire computational procedure is shown in Table 6.9 (principal tableaus 0-3) and in Table 6.10 (auxiliary tableau).

As the initial dual basis we take the same system of vectors as under the first algorithm. The basis variables do not enter the linear form of the problem. The elements in row Δ of the auxiliary tableau therefore coincide with the corresponding linear-form coefficients, taken with the reverse sign. The vector $A_{s_1}=A_8$ corresponding to the greatest (in maximum value) negative pseudoprogram component is eliminated from the basis.

The row X_1, whose entries are computed from formula (6.5), contains some negative values. Hence, case (c) obtains.

In row θ, only the positions 1, 3, and 6 corresponding to negative entries of the row X_1 are filled. The least element of the row θ corresponds to the vector A_3. This vector should be introduced into the first basis position in place of A_8. The expansion coefficients of A_3 in the initial basis are computed from (6.1) and written in column $A_k^{(l)}$ of the principal zeroeth tableau. In the right bottom corner of the tableau we write $\Delta_k=\Delta_3=2$. Then, the main part of the zeroeth tableau is transformed into the main part of the principal 1st tableau by means of the recurrence formulas

$$e'_{1j}=\frac{1}{x_{13}}e_{1j},$$

$$e'_{ij}=e_{ij}-e'_{1j}x_{i3},$$

$$i=1,\ 2,\ 3,\ 4,\ 5,\quad j=0,\ 1,\ 2,\ 3,\ 4,\ 5.$$

For example,

$$e_{11}=\frac{1}{-3}\cdot(-1)=0.333,$$

$$e_{51}=0-0.333\cdot(-2)=0.667.$$

We now proceed with the first stage of the 1st iteration.

The column e_0 again contains negative entries only. The vector $A_{s_4}=A_{11}$ should be eliminated from the basis. The row Δ' of the auxiliary tableau is computed from recurrence formula (6.7). We have, e.g.,

$$\Delta'_6=\Delta_6+\theta_0 x_{16}=2+0.667\cdot(-2)=0.667.$$

The row X'_4 of the auxiliary tableau has some negative entries. Thus, we need not conclude that the problem is unsolvable.

We now proceed with the second stage of the 1st iteration. We fill the row θ'. The least element of this row $\theta'_6=0.222$ is obtained on A_6. The restraint vector A_6 is introduced into the fourth position of the successive basis in place of $A_{s_4}=A_{11}$. In the last column $A_k^{(l)}$ of the principal 1st tableau we write the coefficients x'_{i6} of the vector A_6 in the current basis, as computed from (6.2). The $(m+1)$-th (sixth) entry in column $A_k^{(l)}$ is $\Delta'_6=0.667$.

The principal 1st tableau is, thus, completely filled and ready for the transformation of its main part into the main part of the principal 2nd tableau. Continuing with this procedure we obtain after the third iteration pseudoprogram X''' with positive components. The program X''' solves the problem. The basis components of the optimal support program of the problem are written in column e_0 of the principal 3rd tableau. The maximum of the linear form of the primal problem and the components of the solution vector of the dual problem are written in the last row of the principal 3rd tableau.

In Table 6.11 (principal tableaus 0–2) and in Table 6.12 (auxiliary tableau) we list the sequence of computations under the second dual algorithm for Example 2, considered in 5-3 in connection with the first algorithm.

6-5. Let us now consider some features of the application of the second dual algorithm to linear-programming problems with bilateral restraints.

Taking into consideration the remarks in 5-5, we may, without difficulty, modernize the computational procedure of the second dual algorithm and adapt it to the solution of linear-programming problems with bilateral restraints.

The principal tableaus of the second algorithm for a problem with bilateral restraints differ from the principal tableaus for a problem given in canonical form in having three additional columns, specifically $(\alpha;\ \beta)_X$, δ, and $\tilde{e}_0$ (Table 6.13).

TABLE 6.9 (0–3)

	No.	c_X	B_X	e_0	e_1	e_2	e_3	e_4	e_5	$A_k^{(e)}$	No of tableau
←	1		A_8	−10	−1					−3	0
	2		A_9	−4		−1					
	3		A_{10}	−2			−1			−5	
	4		A_{11}	−5				−1			
	5		A_{12}	−3					−1	−2	
	6	—	L							2	
→	1	−2	A_3	3.333	0.333					0.667	1
	2		A_9	−4		−1				−1	
	3		A_{10}	14.667	1.667		−1			3.333	
←	4		A_{11}	−5				−1		−3	
	5		A_{12}	3.667	0.667				−1	1.333	
	6	—	L	−6.667	−0.667					0.667	
	1	−2	A_3	2.222	0.333			−0.222			2
←	2		A_9	−2.333		−1		0.333		−4	
	3		A_{10}	9.111	1.667		−1	−1.111		−1	
→	4	−2	A_6	1.667				0.333			
	5	—	A_{12}	1.444	0.667			−0.444	−1		
	6		L	−7.778	−0.667			−0.222		1	
	1	−2	A_3	2.222	0.333			−0.222			3
→	2	−1	A_2	0.583		0.25		−0.0833			
	3		A_{10}	9.694	1.667	0.25	−1	−1.194			
	4	−2	A_6	1.667				0.333			
	5		A_{12}	1.444	0.667			−0.444	−1		
	6	—	L	−8.361	−0 667	−0.25		−0.139			

TABLE 6.10

Auxiliary tableau

No.	B	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}
1	10	2		3			2		−1				
2	4		4			3	1			−1			
3	2		1	5		4		5			−1		
4	5	3			2		3					−1	
5	3			2	4			3					−1
6	c	−3	−1	−2	−3	−1	−2	−5					
0	Δ	3	1	2	3	1	2	5					
	X_1	−2		−3			−2		1				
	θ	1.5		**0.667**			1						
1	Δ'	1.667	1		3	1	0.667	5	0.667				
	X_4'	−3			−2		−3					1	
	θ'	0.556	—	—	1.5	—	**0.222**	—	—	—	—	—	—
2	Δ''	1	1		2.556	1		5	0.667			0.222	
	X_2''	1	−4		0.667	−3				1		−0.333	
	θ''	—	**0.25**	—	—	0.333	—	—	—	—	—	0.667	—

The columns $(\alpha;\ \beta)_X$ and δ are filled in the same way as under the first algorithm (see 5-5). The column $\tilde{e}_0$ is related to the vector $e_0 = B - \sum_{j \notin I_\gamma} x_j A_j$ like the vector $\tilde{A}_0$ to A_0 under the first algorithm:

$$\tilde{e}_0 = e_0 + x_k A_k - \gamma_r A_{s_r},$$

or

$$\tilde{e}_{i0} = \begin{cases} e_{i0} + x_k x_{ik} & \text{for } i \neq r, \\ e_{r0} + x_k x_{rk} - \gamma_r & \text{for } i = r. \end{cases}$$

In the upper part of the auxiliary tableau we introduce an $(m+2)$-th row $(\alpha;\ \beta)$ whose entries are the boundary values of the problem variables.

When solving a problem given in canonical form, in each iteration three rows ($\Delta^{(l)}$, $X_r^{(l)}$, and $\theta^{(l)}$) are added in the lower part of the auxiliary tableau (Table 6.14). In a problem with bilateral restraints in each iteration, moreover, it is necessary to fill the row $X^{(l)}$ containing the modifiers α, β of the extra-basis variables of the current pseudoprogram.

The rows $X_r^{(l)}$ and $\Delta^{(l)}$ of the auxiliary tableau are computed following the general rules stated in 6-1 and 6-2 in connection with the second algorithm. The row θ is computed precisely as under the first algorithm for a problem with bilateral restraints (see 5-5).

Let us now briefly outline the sequence of computations in a single $(l+1)$-th iteration.

At the beginning of the $(l+1)$-th iteration we assume to have at our disposal the principal l-th tableau (except the columns $\tilde{e}_0$ and A_k) and the rows of the auxiliary tableau corresponding to the l-th iteration.

Pseudoprogram $X^{(l)}$ is a solution of the problem if all the positions in column δ of the principal l-th tableau are crossed out (case (a)). If the pseudoprogram is not a feasible program of the problem, we proceed with the solution process. The vector A_{s_r} with the largest deviation $\delta_r^{(l)}$ is eliminated from the basis. The rows $X^{(l)}$, $\Delta^{(l)}$, and $X_r^{(l)}$ of the auxiliary tableau are then filled. The row $X^{(l)}$ is filled following the same rules as in the first algorithm (see 5-5). The parameters $\Delta_j^{(l)}$ are computed from recurrence formulas (6.7) using the elements of the preceding rows ($\Delta^{(l-1)}$, $X_r^{(l-1)}$, $\theta^{(l-1)}$). The components of row $X_r^{(l)}$ are computed, as in a problem in canonical form, from (6.5).

Rows $X^{(l)}$ and $X_r^{(l)}$ enable us to perform a limited unsolvability test. (The complete test at this stage necessitates the computation of all $X_r^{(l)}$ for which $x_{i0} < 0$.) Inconsistency of problem restraints is established in one of the following cases:

(i) $\delta_r^{(l)}$ with a modifier α, $x_{rj} \geqslant 0$ for $x_j = \alpha_j$ and $x_{rj} \leqslant 0$ for $x_j = \beta_j$ $(j \notin I_\gamma)$;

(ii) $\delta_r^{(l)}$ with a modifier β, $x_{rj} \leqslant 0$ for $x_j = \alpha_j$ and $x_{rj} \geqslant 0$ for $x_j = \beta_j$ $(j \notin I_\gamma)$.

It is clear that unsolvability, if the problem is actually unsolvable, is established under the second algorithm at a later stage than under the first.

The tests for case (a) and (b) complete the first stage of the iteration. We proceed with the second stage if case (c) obtains. In the second stage we choose the vector A_k to be introduced into the basis and fill the principal $(l+1)$-th tableau (except for columns $\tilde{e}_0$ and A_k).

The computations are made in the following order. The row $\theta^{(l)}$ of the auxiliary tableau is filled according to the rules discussed in connection

TABLE 6.11 (0–2)

	No.	C_X	B_X	e_0	e_1	e_2	e_3	e_4	e_5	$A_k^{(e)}$	No. of tableau
←	1		A_7	−12	−1					−8	0
	2		A_8	−5		−1				−3	
	3		A_9	−7			−1			−1	
	4		A_{10}	−10				−1		−1	
	5		A_{11}	−4					−1	−6	
	6	—	L							1	
→	1	−1	A_4	1.5	0.125					0.125	1
	2		A_8	−0.5	0.375	−1				−0.625	
	3		A_9	−5.5	0.125		−1			−3.875	
←	4		A_{10}	−8.5	0.125			−1		−5.875	
	5		A_{11}	5	0.75				−1	−2.25	
	6	—	L	−1.5	−0.125					1.875	
	1	−1	A_4	1.319	0.128			−0.021			2
	2		A_8	0.404	0.362	−1		0.106			
	3		A_9	0.106	. 0.043		−1	0.660			
→	4	−2	A_5	1.447	−0.021			0.170			
	5		A_{11}	8.255	0.702			0.383	−1		
	6	—	L	−4.213	−0.085			−0.319			

TABLE 6.12

Auxiliary tableau

No.	B	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}
1	12	6	2	3	8	1	2	−1				
2	5	4	5	8	3	1	7		−1			
3	7	2			1	4	2			−1		
4	10	5	2	1	1	6	5				−1	
5	4	1	3		6	3	2					−1
6	C	−5	−3	−4	−1	−2	−3					
0	Δ	5	3	4	1	2	3					
	X_1	−6	−2	−3	−8	−1	−2	1				
	θ	0.833	1.5	1.333	**0.125**	2	1.5	—	—	—	—	—
1	Δ	4.25	2.75	3.625		1.875	2.75	0.125				
	X_4'	−4.25	−1.75	−0.625		−5.875	4.75	−0.125			1	
	θ'	1	1.571	5.8		**0.319**	0.579	1	—	—	—	—

TABLE 6.13

The principal l-th tableau

No.	C_X	B_X	e_0	$\breve{e}_0$	e_1	e_2	$\ldots$	e_m	$(a;\ \beta)_X$	δ	$A_k^{(l)}$
1	c_{s_1}	A_{s_1}	$e_{10}^{(l)}$	$\breve{e}_{10}^{(l)}$	$e_{11}^{(l)}$	$e_{12}^{(l)}$	$\ldots$	$e_{1m}^{(l)}$	$\alpha_{s_1};\ \beta_{s_1}$		$x_{1k}^{(l)}$
2	c_{s_2}	A_{s_2}	$e_{20}^{(l)}$	$\breve{e}_{20}^{(l)}$	$e_{21}^{(l)}$	$e_{22}^{(l)}$	$\ldots$	$e_{2m}^{(l)}$	$\alpha_{s_2};\ \beta_{s_2}$		$x_{2k}^{(l)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ldots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
r	c_{s_r}	A_{s_r}	$e_{r0}^{(l)}$	$\breve{e}_{r0}^{(l)}$	$e_{r1}^{(l)}$	$e_{r2}^{(l)}$	$\ldots$	$e_{rm}^{(l)}$	$\alpha_{s_r};\ \beta_{s_r}$	$\delta_r^{(l)}\ (\gamma_r)$	$x_{rk}^{(l)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ldots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
m	c_{s_m}	A_{s_m}	$e_{m0}^{(l)}$	$\breve{e}_{m0}^{(l)}$	$e_{m1}^{(l)}$	$e_{m2}^{(l)}$	$\ldots$	$e_{mm}^{(l)}$	$\alpha_{s_m};\ \beta_{s_m}$		$x_{mk}^{(l)}$
$m+1$	—	—	$L^{(l)}$	—	$y_1^{(l)}$	$y_2^{(l)}$	$\ldots$	$y_m^{(l)}$	—	—	$\Delta_k^{(l)}$

with the first algorithm (see 5-5). The vector A_k on which

$$\theta_0^{(l)} = \min_j \theta_j^{(l)}$$

is obtained is introduced into the basis. In the degenerate case, as long as no cycling has been detected, the simplified rule for the choice of A_k (from the lowest or the highest index) should be applied. The exact rule guaranteeing against cycling has been stated in 4-4. Observe that this rule requires knowing the coefficients $x_{ij}^{(l)}$ in the expansion of the vectors A_j in terms of the basis of pseudoprogram $X^{(l)}$.

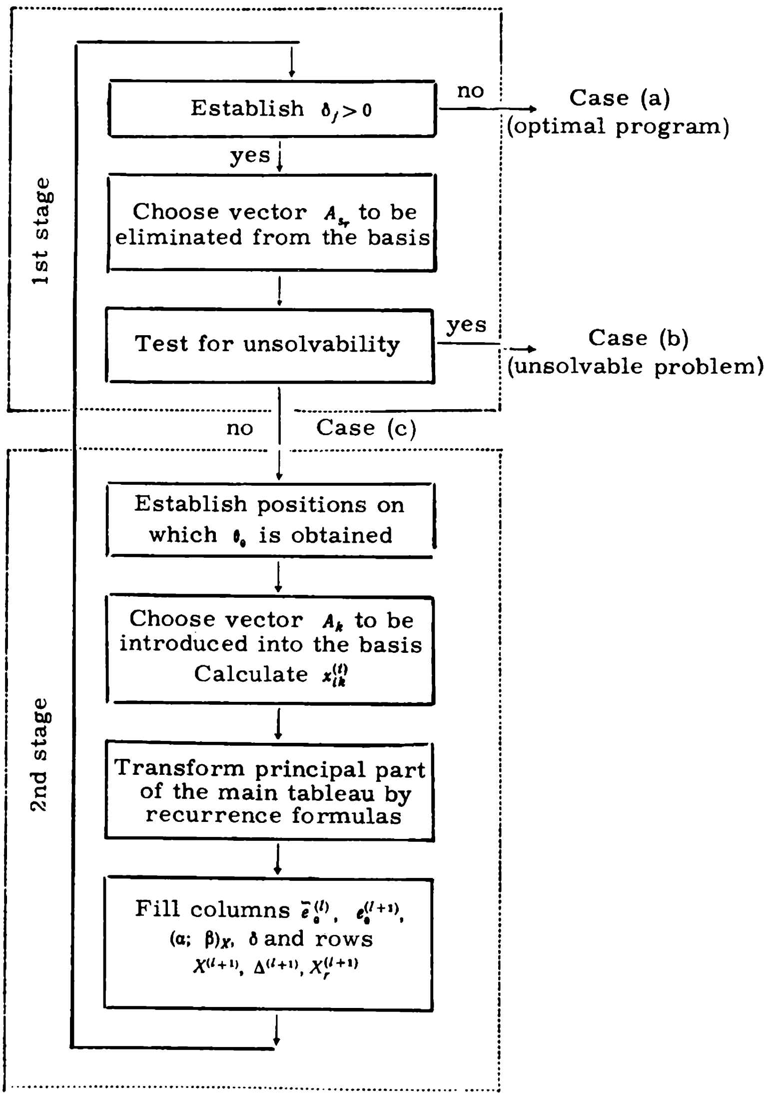


FIGURE 6.8

TABLE 6.14

Auxiliary tableau

No.	B	A_1	A_2	$\dots$	A_k	$\dots$	A_n
1	b_1	a_{11}	a_{12}	$\dots$	a_{1k}	$\dots$	a_{1n}
2	b_2	a_{21}	a_{22}	$\dots$	a_{2k}	$\dots$	a_{2n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$	$\dots$	$\vdots$
m	b_m	a_{m1}	a_{m2}	$\dots$	a_{mk}	$\dots$	a_{mn}
$m+1$	c_j	c_1	c_2	$\dots$	c_k	$\dots$	c_n
$m+2$	$\alpha;\ \beta$	$\alpha_1;\ \beta_1$	$\alpha_2;\ \beta_2$	$\dots$	$\alpha_k;\ \beta_k$	$\dots$	$\alpha_n;\ \beta_n$
0	Δ	Δ_1	Δ_2	$\dots$	Δ_k	$\dots$	Δ_n
	X			$\dots$	γ_k	$\dots$	
	X_r	x_{r1}	x_{r2}	$\dots$	x_{rk}	$\dots$	x_{rn}
	θ			$\dots$	θ_0	$\dots$	
1	Δ'	Δ_1'	Δ_2'	$\dots$	Δ_k'	$\dots$	Δ_n'
	X'			$\dots$	γ_k	$\dots$	
	X_r'	x_{r1}'	x_{r2}'	$\dots$	x_{rk}'	$\dots$	x_{rn}'
	θ'			$\dots$		$\dots$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$	$\dots$	$\vdots$

TABLE 6.15 (0–2)

	No.	C_X	B_X	e_0	$\tilde{e}_0$	e_1	e_2	e_3	e_4	$(\alpha;\beta)$	δ	$A_k^{(e)}$	No. of tableau
←	1		A_5	−8	−2	1				−1; 1	7(α)	5	0
	2	1	A_6	2	5		1			−1; 1	1(β)	3	
	3	2	A_7	−6	−1			1		−1; 1	5(α)	5	
	4		A_8	2	3				1	−1; 1	1(β)	1	
	5	—	Δ	24	—					—	—	−1	
→	1	14	A_4	−0.4	−0.6	0.2				−1; 1	—	0.2	1
←	2	1	A_6	6.2	1.8	−0.6	1			−1; 1	5.2(β)	3.4	
	3	2	A_7	1		−1		1		−1; 1	—	1	
	4		A_8	3.4	1.6	−0.2			1	−1; 1	2.4(β)	1.8	
	5	—	Δ	22.6	—	0.2				—	—	6.2	
	1	14	A_4	−0.706		0.235	−0.059			−1; 1			2
→	2	2	A_2	0.529		−0.176	0.294			−1; 1			
	3	2	A_7	−0.529		−0.824	−0.294	1		−1; 1			
	4	—	A_8	0.647		0.118	−0.529		1	−1; 1			
	5	—	Δ	13.118	—	1.294	−1.824			—	—		

The parameters x_{ij} appear in the first-algorithm tableaus. In the second algorithm we compute (from (6.2)) only the coefficients $x_{ik}^{(l)}$ — the elements of the last column $A_k^{(l)}$ of the principal l-th tableau; and $x_{rj}^{(l)}$ — the elements of the row $X_r^{(l)}$ of the auxiliary tableau. When the vector A_k to be introduced into the basis is uniquely chosen following the exact rule, the coefficients $x_{ij}^{(l)}$ must be computed in the degenerate case whenever neccesary from formulas (6.2). The application of the rule guaranteeing against cycling in the second algorithm involves considerably more complicated computations than in the first.

TABLE 6.16

No.		B	A_1	A_2	A_3	$\bar{A}_4$	A_5	A_6	A_7	A_8
1		1	2	1	3	5	1			
2		4	1	4	2	3		1		
3		4	4	2	3	5			1	
4		8	3	2	4	1				1
5		c_j	12	2	10	14		1	2	
6		$\alpha;\beta$	$-1;1$	$-1;1$	$-1;1$	$-1;1$	$-1;1$	$-1;1$	$-1;1$	$-1;1$
0	Δ		-3	6	-2	-1				
	X		β	α	β	β	—	—	—	—
	X_r		2	1	3	5	1			
	θ		1.5	—	0.667	**0.2**	—	—	—	—
1	Δ'		-2.6	6.2	-1.4		0.2			
	X'		β	α	β	—	α	—	—	—
	X'_r		-0.2	3.4	0.2		-0.6			
	θ'		13	**1.824**	—	—	—	—	—	—
2	Δ''		-2.235		-1.765		1.294	-1.824		
	X''		β	—	β	—	α	β	—	—
	X''_r									
	θ''			—		—			—	—

Having determined the elements $x_{ik}^{(l)}$ of the last column in the principal l-th tableau, we proceed to fill the main part of the principal $(l+1)$-th tableau. The columns $e_j (j=1,\ 2,\ \ldots,\ m)$ of the $(l+1)$-th tableau are expressed in terms of the elements of the l-th tableau using recurrence formulas (6.3). The component of the successive pseudoprogram, i.e., the entries of column e_0 in the $(l+1)$-th tableau are determined using the same recurrence formulas from the entries in column $\bar{e}_0$ of the l-th tableau:

$$e_{i0}^{(l+1)} = \begin{cases} \bar{e}_{i0}^{(l)} - \dfrac{x_{ik}^{(l)}}{x_{rk}^{(l)}} \bar{e}_{r0}^{(l)} & \text{for } i \neq r, \\ \dfrac{\bar{e}_{r0}^{(l)}}{x_{rk}^{(l)}} & \text{for } i = r. \end{cases}$$

The columns $(\alpha;\beta)_X$ and δ are transformed as in the first algorithm. The value of the linear form is transformed between two successive iterations according to formula (5.17).

The $(m+1)$-th row of the principal tableau is required only to compute the initial values of the parameters Δ. In subsequent stages the kernel components of the successive support program of the dual problem are required for control purposes only. The components $y_j = e_{m+1}, _j \, (j=1, 2, \ldots, m)$ can be computed either directly from (6.1), or from recurrence formulas (6.3). Another possibility is to determine the parameters Δ_j simultaneously from recurrence formulas (6.7) and from the defining relationships (6.4).

This sequence of computations produces after a finite number of iterations either case (a) or case (b).

In Figure 6.8 we show a block diagram of the solution of a linear-programming problem with bilateral restraints according to the second dual algorithm.

In Table 6.15 (principal tableaus 0–2) and Table 6.16 (auxiliary tableau) we give the process of solution according to the second dual algorithm of the linear-programming problem with bilateral restraints considered in 5-5 in connection with the first algorithm.

§ 7. Methods of determining the initial support program of the dual problem

7-1. The solution of a linear-programming problem by the dual method sets out from a known dual basis.

In some cases the concrete form of the restraints or the physical meaning of the problem makes it possible to establish an initial support program of the dual problem, and consequently the dual basis, easily. Among such problems, as we have seen in 3-7, are linear-programming problems with bilateral restraints. In 7-2 we shall consider another class of problems where an initial dual basis can be found directly.

In other cases we can easily determine an initial feasible program of the dual problem, without being sure that this is a support program. Examples of such problems are given in 7-2. We should, naturally, expect that the construction of an initial support program is facilitated if a feasible program of the problem is known. In 7-3 we describe a method; and in 7-4 an algorithm for constructing a support program from any feasible program. In 7-5 and 7-6 the method described is illustrated by examples.

In 7-7 and 7-8 we consider two ways for solving a problem by the dual method in cases when the determination of an initial, even feasible, program involves some difficulties.

7-2. Consider a class of linear-programming problems where the initial dual basis can be determined without computation.

Let the linear-programming problem be given in the following form:

Maximize the linear form

$$L(X) = \sum_{j=1}^{n} c_j x_j \tag{7.1}$$

subject to the conditions

$$\sum_{j=1}^{n} a_{ij} x_j \geqslant b_i, \qquad i = 1, 2, \ldots, m, \tag{7.2}$$

$$x_j \geqslant 0, \qquad j = 1, 2, \ldots, n. \tag{7.3}$$

Now let all the coefficients c_j ($j=1, 2, \ldots, n$) of linear form (7.1) be nonpositive.

The problem reduces to canonical form when restraints (7.2), (7.3) are replaced by

$$\sum_{j=1}^{n} a_{ij}x_j - x_{n+i} = b_i, \quad i=1, 2, \ldots, m, \tag{7.4}$$

$$x_j \geqslant 0, \quad j=1, 2, \ldots, n+m. \tag{7.5}$$

The dual problem of problem (7.1), (7.4), (7.5) is stated as follows: Minimize the linear form

$$\bar{L}(Y) = \sum_{i=1}^{m} b_i y_i \tag{7.6}$$

subject to the conditions

$$\sum_{i=1}^{m} a_{ij} y_i \geqslant c_j, \quad j=1, 2, \ldots, n, \tag{7.7}$$

$$-y_i \geqslant 0, \quad i=1, 2, \ldots, m. \tag{7.8}$$

The vector

$$Y = \underbrace{(0, \ldots, 0)}_{m}$$

is obviously a support program of the dual problem (7.6)–(7.8). If all $c_j<0$, the support program Y is also nondegenerate. The dual basis thus comprises the vectors

$$A_{n+i} = \underbrace{(\overbrace{0, \ldots, 0, -1}^{i}, 0, \ldots, 0)}_{m}.$$

The vectors A_{n+i} ($i=1, 2, \ldots, m$) thus constitute the basis of the initial pseudoprogram. The basis components of the initial pseudoprogram are equal to minus the corresponding components of the constraint vector. Similarly, the coefficients x_{ij} in the expansion of the restraint vectors A_j in terms of the initial pseudoprogram basis are equal to minus the corresponding components of these vectors.

Now consider a class of problems where the initial support program cannot be indicated that easily, yet a feasible program can be established without difficulty.

Let in problem (1.1)–(1.3) a certain fixed component a_{tj} of all the restraint vectors A_j be positive. Consider the vector

$$Y = \underbrace{(\overbrace{0, \ldots, 0, y_t}^{t}, 0, \ldots, 0)}_{m},$$

where

$$y_t \geqslant \max_{1 \leqslant j \leqslant n} \frac{c_j}{a_{tj}}.$$

The vector Y satisfies all the restraints of problem (1.4)–(1.5) and is, therefore, a feasible program of the dual problem. Vector Y, unfortunately, is generally not a support program. It is, therefore, interesting to study the method enabling us to obtain a support program from a known feasible program.

7-3. The method of determining a support program discussed in the following is called the g r a d i e n t m e t h o d. We first consider, in geometrical concepts, the essentials of this method.

The system of inequality restraints of the dual problem defines in the m-dimensional space of the variables $y_1, y_2, \ldots, y_m$ a polyhedral set S, the set of feasible programs of the dual problem. Each support program corresponds to a vertex of set S.

Let some feasible program Y', which does not coincide with any of the vertices of S, be given. The point Y' lies inside some face S_1 of the polyhedral set S (S_1 may coincide with S). The linear form of the dual problem generates on S_1 some linear function. The gradient of this function is parallel to the projection of the constraint vector B on the face S_1. We now let the point Y' move in the direction of fastest decrease of the linear function, until it meets the boundary S'. The new program Y'' lies inside the face S_2, whose dimensionality is strictly less than that of S_1. Projecting the direction vector of the linear-form hyperplane $\tilde{L}$ onto the face S_2, we obtain the direction of fastest decrease of the linear function defined by $\tilde{L}$ on S_2.

Proceeding in this way and lowering in each step the dimensionality of the corresponding faces, we finally arrive at a zero-dimensional face, i.e., a vertex of the polyhedral set S. This vertex corresponds to a support program of the dual problem. This support program is, as a rule, a fairly satisfactory approximation to the solution of the problem. This is quite obvious: we moved from Y' to Y'' and from Y'' to Y''', etc., in the direction of fastest decrease of the linear form $\tilde{L}$ on the corresponding faces of set S.

The gradient method may lead to a face S_p parallel to the linear-form hyperplane. The linear form retains a constant value on this face. Further motion is possible, generally speaking, in any direction, as long as it leads to the boundary of this face. If the face S_p is a polyhedron, rectilinear motion in any direction will lead to a program image on a face of lower dimensionality. The choice of the direction of motion of the point $Y^{(p)}$ is somewhat limited, if S_p is an unbounded polyhedral set. It is, nevertheless, possible to reach the boundary of the face in question.

We will now interpret these geometrical considerations analytically.

Let $Y' = (y'_1, \ldots, y'_m)$ be a feasible program of the dual problem. Without loss of generality, we may take

$$\sum_{i=1}^{m} a_{ij} y'_i > c_j, \qquad j = 1, 2, \ldots, n_1, \tag{7.9}$$

$$\sum_{i=1}^{m} a_{ij} y'_i = c_j, \qquad j = n_1 + 1, \ldots, n. \tag{7.10}$$

In particular, n_1 may be equal to n.

Now, among equalities (7.10) let there be r linearly independent equations. If $r = m$, Y' is a support program of the problem. Let $r < m$.

Consider the system of equalities and inequalities specified by program Y':

$$\sum_{i=1}^{m} a_{ij} y_i \geqslant c_j, \qquad j = 1, 2, \ldots, n_1, \tag{7.11}$$

$$\sum_{i=1}^{m} a_{ij} y_i = c_j, \qquad j = n_1 + 1, \ldots, n. \tag{7.12}$$

Relationships (7.11), (7.12) define a face S_1 of the polyhedral restraint set S of the dual problem; the set S_1 is the face with minimum dimensionality which still contains the point Y'.

We express r variables (to be specific, $y_1, y_2, \ldots, y_r$) from (7.12) in terms

of the remaining m-r variables

$$y_j=\sum_{i=r+1}^{m} d_{ij}y_i+d_j, \qquad j=1,\ 2,\ \ldots,\ r. \tag{7.13}$$

We now insert the expressions for $y_i\,(i=1,2,\ldots,r)$ from (7.13) into (7.11) and obtain a system of inequalities for $y_{r+1}, y_{r+2}, \ldots, y_m$:

$$\sum_{i=r+1}^{m} a'_{ij}y_i\geqslant c'_j, \qquad j=1,\ 2,\ \ldots,\ n_1. \tag{7.14}$$

System (7.14) obviously defines the necessary and sufficient conditions for the vector $Y=(y_1, \ldots, y_m)$, whose components satisfy equalities (7.12) (or (7.13)), to be a feasible program of problem (1.4)–(1.5).

We now insert the expressions for $y_1, y_2, \ldots, y_r$ from (7.13) into (1.4) and obtain a linear function of the variables $y_{r+1}, \ldots, y_m$:

$$\tilde{L}'=\sum_{i=r+1}^{m} b'_iy_i+b_0. \tag{7.15}$$

Let us now move the point $Y'=(y'_1, \ldots, y'_m)$in the direction of fastest decrease of the function (7.15) until at least one of the inequalities (7.14) reduces to an equality. The direction of fastest decrease of $\tilde{L}'$ is defined by minus its gradient vector

$$-B'=-(b'_{r+1}, \ldots, b'_m).$$

It is assumed that linear form (7.15) does not retain a constant value on face S_1. In other words, some of the coefficients b_i, $i=r+1, \ldots, m$ are nonzero. Let the new program be denoted by Y'' The point Y'' lies in a face of lower dimensionality than the dimensionality of the face in which Y' lies.

Analytically, motion in the direction of fastest decrease indicates transformation from Y' to $Y'(\theta)$, where

$$y'_i(\theta)=y'_i-\theta b'_i, \qquad i=r+1,\ r+2,\ \ldots,\ m,\ \theta>0.$$

The remaining components of $Y'(\theta)$ are determined from (7.12) or (7.13).

Let us now compute the parameters $\Delta_j(\theta)$ for $Y'(\theta)$:

$$\Delta_j(\theta)=\sum_{i=r+1}^{m} a'_{ij}y'_i(\theta)-c_j=\sum_{i=r+1}^{m} a'_{ij}(y'_i-\theta b'_i)-c'_j=$$
$$=\Delta_j-\theta\sum_{i=r+1}^{m} a'_{ij}b'_i \qquad (j=1,\ 2,\ \ldots,\ n).$$

Let

$$\sum_{i=r+1}^{m} a'_{ij}b'_i=\mu_j, \tag{7.16}$$

whence

$$\Delta_j(\theta)=\Delta_j-\theta\mu_j. \tag{7.17}$$

The point $Y'(\theta)$ belongs to the polyhedral set defined by conditions (7.11), (7.12), i. e., the face S_1 of the set S, for all θ satisfying conditions of the form (7.14), or, equivalently,

$$\Delta_j(\theta)\geqslant 0, \qquad j=1,\ 2,\ \ldots,\ n_1.$$

From (7.17), we obtain the domain of definition of θ:

$$\Delta_j\geqslant\theta\mu_j, \qquad j=1,\ 2,\ \ldots,\ n. \tag{7.18}$$

According to (7.9), $\Delta_j>0$ for $j=1,\ 2,\ \ldots,\ n_1$. Therefore, if all $\mu_j\leqslant 0$,

we may take θ arbitrarily large and $Y'(\theta)$ will not be outside the domain of definition of $\tilde{L}'$. This indicates that $\tilde{L}'$ for $\mu_j \leqslant 0$ $(j=1, 2, \ldots, n_1)$ is unbounded below in the set of the variables $y_{r+1}, \ldots, y_m$ satisfying (7.14). In other words, for $\mu_j \leqslant 0 (j=1, 2, \ldots, n_1)$ the dual problem is unsolvable.

Now let some of the μ_j be positive. Taking

$$\theta_0 = \min_{\mu_i > 0} \frac{\Delta_j}{\mu_j}, \tag{7.19}$$

we obtain program $Y'' = (y_1'', y_2'', \ldots, y_m'')$, where $y_i'' = y_i'(\theta_0)$ for $i=r+1, \ldots, m$, and the components $y_1'', \ldots, y_r''$ are expressed from (7.13) in terms of the other known variables.

From the definition of Y'',

$$\sum_{i=1}^{m} a_{ij} y_i'' = c_j, \qquad j = n_1 + 1, \ldots, n.$$

Let θ_0 be obtained on $j=k$,

$$\theta_0 = \min_{\mu_j > 0} \frac{\Delta_j}{\mu_j} = \frac{\Delta_k}{\mu_k}.$$

Hence, $\Delta_k(\theta_0) = 0$, or, equivalently,

$$\sum_{i=1}^{m} a_{ik} y_i'' = c_k. \tag{7.20}$$

The equality

$$\sum_{i=1}^{m} a_{ik} y_i = c_k \tag{7.21}$$

is linearly independent of system (7.12). Indeed, system (7.12) is equivalent to (7.13). Hence, (7.21) can be written as

$$\sum_{i=r+1}^{m} a'_{ik} y_i = c'_k. \tag{7.22}$$

From (7.16), $\mu_k > 0$ implies that at least one of the a'_{ik} is nonzero. Let $a'_{sk} \neq 0$ and the matrix of coefficients D of conditions (7.13), (7.22) have the form

$$D = \left\| \begin{array}{cccccccc} 1 & 0 & \ldots & 0 & -d_{r+1,1} & \ldots & -d_{m1} \\ 0 & 1 & \ldots & 0 & -d_{r+1,2} & \ldots & -d_{m2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \ldots & 1 & -d_{r+1,r} & \ldots & -d_{mr} \\ 0 & 0 & \ldots & 0 & a'_{r+1,k} & \ldots & a'_{mk} \end{array} \right\|.$$

The determinant of the submatrix comprising the first r columns of the matrix D and the column containing the element a'_{sk} is equal to a'_{sk} and is thus nonzero. This indicates that (7.22) is linearly independent of the system (7.12).

Program Y'' thus reduces at least $r+1$ linearly independent inequalities from among the restraints of the dual problem to equalities. Here

$$\tilde{L}(Y'') = \sum_{i=1}^{m} b_i y_i'' = \sum_{i=r+1}^{m} b_i' y_i'(\theta_0) + b_0' =$$

$$= \sum_{i=r+1}^{m} b_i' y_i' + b_0' - \theta_0 \sum_{i=r+1}^{m} (b_i')^2 < \sum_{i=r+1}^{m} b_i' y_i' + b_0' = \sum_{i=1}^{m} b_i y_i' = \tilde{L}(Y').$$

Thus, if at least one of the coefficients b_i' $(i=r+1, \ldots, m)$ is nonzero, each step of the gradient method decreases the linear form of the dual problem.

Now let all b_i' be zero. Geometrically, this indicates that the face S_1, containing the program Y' of the problem, is parallel to the linear-form hyperplane and, consequently, L' retains a constant value (equal to b_0') on S_1. In this case the boundary of S_1 can be attained if Y' is moved as follows

$$\left.\begin{array}{l} y''_{r+1}(\theta) = y'_{r+1} - \theta, \\ y''_i = y'_i, \quad i = r+2, \ldots, m. \end{array}\right\} \tag{7.23}$$

(Any of the y_i' with $i > r$ can, obviously, play the role of the variable y_{r+1}.)

The conditions for the domain of definition of θ are

$$\Delta_j(\theta) = \sum_{i=r+1}^{m} a'_{ij} y'_i(\theta) - c'_j \geqslant 0, \qquad j = 1, 2, \ldots, n_1,$$

or, equivalently,

$$\Delta_j(\theta) = \Delta_j - \theta a'_{r+1,\,j} \geqslant 0, \qquad j = 1, 2, \ldots, n_1. \tag{7.24}$$

All $a'_{r+1,\,j}$ cannot vanish simultaneously. Otherwise, no $m-r$ linearly independent inequalities could be chosen from (7.11) and the system of restraints (1.5) would contain less than m independent relationships.

The parameter θ_0, i. e., the limiting value of θ for which we remain in the face S_1, is determined, according to (7.24), as follows:

$$\theta_0 = \begin{cases} \min\limits_{a'_{r+1,j} > 0} \dfrac{\Delta_j}{a'_{r+1,\,j}}, & \text{if some } a'_{r+1,\,j} \text{ are positive} \\ -\min\limits_{a'_{r+1,j} < 0} \left(-\dfrac{\Delta_j}{a'_{r+1,\,j}}\right), & \text{if } a'_{r+1,\,j} \leqslant 0 \text{ for } j = 1, 2, \ldots, n_1. \end{cases} \tag{7.25}$$

Observe that for $a'_{r+1,\,j} \leqslant 0$ $(j = 1, 2, \ldots, n_1)$ θ_0 is negative. The number of linearly independent restraints reduced by $Y'' = Y'(\theta_0)$ to equalities increases. The linear form, however, retains its previous value. Proceeding with this technique we, necessarily, obtain a support program of the dual problem having the same value of the linear form. This, obviously, requires at most m steps.

The gradient method for constructing a support program from a given feasible program was discussed here with reference to the dual problem, because it is generally easier to determine a feasible program of the dual problem (1.4)–(1.5) than a feasible program of the primal problem (1.1)–(1.3). If, however, a feasible program of the primal problem is known, the same gradient method will produce a support program of the primal problem. Only in this case it is advisable to modify the computational procedure so that it allows for the specific features of linear-programming problems in canonical form.

7-4. Computations by the gradient method can be made quite compact. The transition from one face of the polyhedral restraint set to another face of lower dimensionality involves solving a system of linear equations by the method of successive elimination of variables. The computational procedure of the gradient method can be represented in a succession of tableaus. The initial zeroeth tableau is shown in Table 6.17. Here

$$b_i = a_{i0}, \qquad i = 1, 2, \ldots, m,$$
$$c_j = a_{m+1,\,j}, \qquad j = 1, 2, \ldots, n,$$
$$a_{m+1,\,0} = b_{m+1} = c_0 = L(X).$$

In the columns of the tableau (in the first $m+1$ positions) we have the

components of the augmented constraint vector $\bar{A}_0$ and the augmented restraint vectors $\bar{A}_j$:

$$\bar{A}_0=\bar{B}=(b_1, \ldots, b_m, L(X))=(a_{10}, \ldots, a_{m0}, a_{m+1,0}),$$
$$\bar{A}_j=(a_{1j}, \ldots, a_{mj}, c_j)=(a_{1j}, \ldots, a_{mj}, a_{m+1,j});$$
$$j=1, 2, \ldots, n.$$

We shall call the matrix $\|a_{ij}\|$ $(i=1, 2, \ldots, m, m+1; j=0, 1, 2, \ldots, n)$ the main part of the tableau.

TABLE 6.17

Zeroeth tableau

№	$\bar{B}=\bar{A}_0$	$\bar{A}_1$	$\bar{A}_2$	...	$\bar{A}_n$
1	a_{10}	a_{11}	a_{12}	...	a_{1n}
2	a_{20}	a_{21}	a_{22}	...	a_{2n}
⋮	⋮	⋮	⋮		⋮
m	a_{m0}	a_{m1}	a_{m2}	...	a_{mn}
$m+1$	$L(X)$	$a_{m+1,1}$	$a_{m+1,2}$	...	$a_{m+1,n}$
$m+2$	—	Δ_1	Δ_2	...	Δ_n

In the last $(m+2)$-th row of the tableau we write the differences (Δ) between the left- and the right-hand sides of restraints (1.5) obtained when some feasible program Y' of the problem is inserted for Y. All the elements in this row are nonnegative.

The position Δ_0 is left empty.

Let the elements of the $(m+2)$-th row $\Delta_{t_1}, \ldots, \Delta_{t_{n-n_1}}$ vanish for $Y=Y'$. Program Y' thus reduces $n-n_1$ restraints in (1.5) to equalities. We apply these equations to eliminate part of the variables from the other restraints in (1.5) and from the linear form. It is convenient to eliminate the variables successively, one by one, transforming the system using the recurrence formulas of the Gauss complete reduction method.

We now use the t_1-th equation to express one of the variables, say y_k (the only requirement imposed on the index $k \leqslant m$ is that $a_{kt_1} \neq 0$) and insert the result into the other restraints and into the linear form. The main part of the zeroeth tableau (the matrix $\|a_{ij}\|$) is thus transformed into the main part of the 1st tableau (the matrix $\|a'_{ij}\|$). The main parts of two successive tableaus are related by the recurrence formulas

$$a'_{ij}=\begin{cases} a_{ij}-\dfrac{a_{kj}}{a_{kt_1}}\, a_{it_1} & \text{for } j \neq t_1, \\ \dfrac{a_{it_1}}{a_{kt_1}} & \text{for } j=t_1, \end{cases} \tag{7.26}$$
$$i=1, 2, \ldots, m, m+1; \; j=0, 1, 2, \ldots, n$$

or, equivalently,

$$a'_{ij}=\begin{cases} a_{ij}-a'_{it_1}\cdot a_{kj} & \text{for } j\neq t_1,\\ \dfrac{a_{it_1}}{a_{kt_1}} & \text{for } j=t_1,\end{cases} \tag{7.26'}$$
$$i=1,\ 2,\ \ldots,\ m,\ m+1;\ j=0,\ 1,\ 2,\ \ldots,\ n.$$

We see from formulas (7.26) that the k-th row of the 1st tableau reduces to the unit row vector

$$a'_{kj}=\begin{cases} 0 & \text{for } j\neq t_1,\\ 1 & \text{for } j=t_1.\end{cases}$$

We shall call these unit row vectors transformed rows.

If $n-n_1>1$, the $(m+2)$-th row of the zeroeth tableau is not transformed ($\Delta'=\Delta$).

In the same way we pass from the 1st tableau to the 2nd tableau, etc., until all the $n-n_1$ have been used to eliminate the variables and transform the matrix $\|a_{ij}\|$ $(i=1,\ \ldots,\ m+1;\ j=0,\ 1,\ \ldots,\ n)$.Consequently, we obtain a tableau with r transformed rows, i. e., $(n+1)$-dimensional unit vectors. The unit elements of these rows are written in columns corresponding to zero Δ_j. The process is terminated when zero elements stand in all the intersections of the columns $\Delta_j=0$ and nontransformed rows.

For $n=n_1$ there is no need for all these transformations.

If $r=m$, the problem is solved, i. e., the vectors A_j corresponding to the units of the transformed rows constitute the basis of a support program of the dual problem.

Now let $r<m$. In all the tableaus, except the last, the $(m+2)$-th row is not transformed, $\Delta=\Delta'=\Delta''=\ldots$ In the last tableau, the one with r transformed rows, we should transform the row Δ into the row $\Delta^{(r)}=\Delta(\theta_0)$ according to (see (7.17) and (7.24))

$$\Delta_j^{(r)}=\Delta_j(\theta_0)=\Delta_j-\theta_0^{(r)}\mu_j^{(r)}, \tag{7.27}$$

where

$$\mu_j^{(r)}=\begin{cases} \sum\limits_{i=r+1}^{m} a_{s_ij}^{(r)}a_{s_i0}^{(r)}, & \text{if at least one}\\ \qquad a_{s_i0}^{(r)}=b_{s_i}^{(r)} & \text{is nonzero } (i=r+1,\ldots m),\\ a_{s_{r+1}j}, & \text{if all } b_{s_i}^{(r)} \text{ are zero } (i=r+1,\ldots,m). \end{cases} \tag{7.28}$$

Here s_i are the indices of the nontransformed rows, j corresponds to columns with $\Delta_j>0$,

$$\theta_0^{(r)}=\begin{cases} \min\limits_{\mu_j>0}\dfrac{\Delta_j}{\mu_j^{(r)}}, & \text{if some of the } \mu_j^{(r)} \text{ are positive,}\\ -\min\limits_{\mu_j<0}\left(-\dfrac{\Delta_j}{\mu_j^{(r)}}\right), & \text{if all } b_{s_i}^{(r)} \text{ are zero and all } \mu_j^{(r)}\leqslant 0. \end{cases} \tag{7.29}$$

In the expression for $\mu_j^{(r)}$ the subscript s_{r+1} is the index of any of the nontransformed rows where all the elements corresponding to $\Delta_j>0$ are zero.

In the tableaus corresponding to the steps of the gradient method in which Δ is transformed it is advisable to write the parameters $\Delta_j^{(r)}$ not in the $(m+2)$-th row (as in the zeroeth tableau), but in the $(m+4)$-th row. In these tableaus $\mu_j^{(r)}$ are written in the $(m+2)$-th row, and $\theta^{(r)}$, i. e., the values of the ratio $\frac{\Delta_j}{\mu_j^{(r)}}$, in the $(m+3)$-th row. The elements of the row $\theta^{(r)}$ are

computed only for those positions where $\mu_j^{(r)} > 0$. The parameters $\mu_j^{(r)}$, in turn, are computed only for $\Delta_j > 0$. In cases when all $b_{si}^{(r)}$ are zero and all $\mu_j^{(r)} \leqslant 0$, entries should be made in the positions of the row $\theta^{(r)}$ corresponding to negative $\mu_j^{(r)}$.

The problem is unsolvable if some of the $b_{si}^{(r)}$ are nonzero and yet all $\mu_j \leqslant 0$.

If there is no indication of unsolvability of the problem, the gradient method should be resumed. The transformation of the row Δ will introduce one or several additional zeros into the row $\Delta^{(r)}$ of the r-th tableau.

The main part of the r-th tableau is again transformed using recurrence formulas (7.26) until the corresponding number of variables have been eliminated from the restraints of the dual problem. Recurrence formula (7.27) is then applied to transform Δ.

The process is continued until all the m first rows of the tableau are reduced to unit vectors or until unsolvability of the problem is established. The restraint vectors corresponding to units in the transformed rows constitute the required basis of the dual problem.

Until now all the rows of the tableau were assumed linearly independent. We will now drop this assumption.

Let the computations connected with one of the tableaus (say, the p-th tableau) produce a row (say the l-th row) whose nonzero elements are located only in the columns containing the units of the transformed rows. Then, one of the following two cases may arise, depending on whether the l-th element in column A_0 is zero or nonzero:

(i) the rank of the restraint matrix $\|a_{ij}\|$ $(i = 1, 2, \ldots, m; j = 1, 2, \ldots, n)$ is less than m, if $a_{l_0}^{(p)} = b_l^{(p)} = 0$;

(ii) the problem is unsolvable, if $a_{l_0}^{(p)} = b_l^{(p)} \neq 0$.

Indeed, by assumption, the l-th row of the p-th tableau is a linear combination of the transformed rows:

$$a_{lj}^{(p)} = \sum_{\substack{i=1 \\ i \neq l}}^{m} \alpha_i a_{ij}^{(p)} \tag{7.30}$$

($\alpha_i = 0$ for the indices corresponding to nontransformed rows).

The equivalent dual problem obtained from the transformations which produced the p-th tableau is written in the following form:

Minimize the linear form

$$\tilde{L}^{(p)}(Y) = \sum_{i=1}^{m} b_i^{(p)} y_i$$

subject to the conditions

$$\sum_{i=1}^{m} a_{ij}^{(p)} y_i \geqslant c_j^{(p)}.$$

Following (7.30) we rewrite the restraints of the equivalent dual problem in the form

$$\sum_{\substack{i=1 \\ i \neq l}}^{m} a_{ij}^{(p)} (y_i + \alpha_i y_l) \geqslant c_j.$$

Let $y' = (y_1', \ldots, y_l', \ldots, y_m')$ be a feasible program of the dual problem. Let

$$y_i^* = y_i' + \alpha_i y_l' \quad \text{for } i \neq l.$$

The vector

$$y = (y_1, \ldots, y_l, \ldots, y_m),$$

where

$$y_i + \alpha_i y_l = y_i^{*}, \qquad i \neq l, \tag{7.31}$$

is obviously a feasible program of the dual problem. Clearly, for any y_l there exist y_i for which (7.31) are satisfied.

Let us now compute the value of the linear form $\tilde{L}^{(p)}$ on program Y:

$$\tilde{L}^{(p)}(Y) = \sum_{\substack{i=1\\ i\neq l}}^{m} b_i^{(p)}(y_i^{*} - \alpha_i y_l) + b_l^{(p)} y_l = \sum_{\substack{i=1\\ i\neq l}} b_i^{(p)} y_i^{*} + y_l\left(b_l^{(p)} - \sum_{\substack{i=1\\ i\neq l}}^{m} \alpha_i b_i^{(p)}\right). \tag{7.32}$$

First, consider the case when

$$b_l^{(p)} = \sum_{\substack{i=1\\ i\neq l}}^{m} \alpha_i b_i,$$

i. e., when $b_l^{(p)} = a_{l_0}^{(p)}$ is the same linear combination of the corresponding parameters of the transformed rows as the other entries of the l-th row. In this case the entire l-th row of the p-th tableau is a linear combination of other rows. Transformation of the tableau does not change the rank of $\|a_{ij}\|$ $(i = 1, 2, \ldots, m, m+1;\ j = 0, 1, \ldots, n)$.

Thus, we reach the conclusion that some of the problem restraints are linearly dependent and the rank of the restraint matrix is thus less than m (case (i)). The l-th row of the p-th tableau is crossed out in this case and left out in future computations.

Now let $b_l^{(p)} \neq \sum\limits_{i=1\, i\neq l} \alpha_i b_i^{(p)}$. We have already seen that there exist feasible programs satisfying (7.31) with an arbitrary y_l. Therefore, if the coefficient of y_l in (7.32) is nonzero, $\tilde{L}^{(p)}$ and, consequently, linear form $\tilde{L}$ of the initial dual problem is unbounded in the set of feasible programs (case (ii)).

One more remark on the algorithm of the gradient method.

The coefficients $c_j = a_{m+1,j}$ are required to compute Δ_j in the zeroeth tableau (in subsequent tableaus, Δ_j either does not change, or is transformed according to recurrence formulas) and to compute the components of the support program of the dual problem. It is often necessary, however, to determine not a support program of the dual problem, but rather the corresponding dual basis. In these cases the $(m+1)$-th row can be omitted from all tableaus, except the zeroeth tableau.

7-5. We now illustrate the application of the gradient method by an appropriate example.

Example 1. Minimize the linear form

$$\tilde{L}(Y) = 2y_1 + 3y_2 + 3y_3 \tag{7.33}$$

subject to the conditions

$$\left.\begin{aligned} y_1 + 2y_2 + y_3 &\geq 1,\\ 2y_1 + y_2 + 2y_3 &\geq 1,\\ y_1 + y_2 + 2y_3 &\geq 1,\\ 2y_1 + 2y_2 + 3y_3 &\geq 2,\\ 2y_1 + 2y_2 + 2y_3 &\geq 1. \end{aligned}\right\} \tag{7.34}$$

Solution. Here $m = 3$, $n = 5$. It can easily be verified that the vector $y = (0, 0, 1)$ is a feasible program of problem (7.33)–(7.34).

The entire computational procedure by which the support program is determined is shown in Table 6.18 (0–3). In the zeroeth tableau we write the problem restraints. In the $(m+2)$-th (fifth) row we indicate the values of Δ, i.e., the differences between the left- and the right-hand sides of restraints (7.34) for $Y = (0, 0, 1)$.

TABLE 6.18 (0–3)

No.	$\overline{A}_0$	↓ $\overline{A}_1$	$\overline{A}_2$	↓ $\overline{A}_3$	↓ $\overline{A}_4$	$\overline{A}_5$	No. of tableau
1	2	1	2	1	2	2	
2	3	2	1	1	2	2	
3	3	**1**	2	2	3	2	0
4	C	1	1	1	2	1	
5	Δ	×	1	1	1	1	
1	−1	1		−1	**−1**		
2	−3	2	−3	−3	−4	−2	
3		1					
4	C'	1	−1	−1	−1	−1	1
5	μ'	×	9	10	13	6	
6	θ'	×	0.111	0.1	**0.0769**	0.167	
7	Δ'	×	0.308	0.231	×	0.538	
1					1		
2	1	−2	−3	**1**	4	−2	
3		1					
4	C''		−1		1	−1	2
5	μ''	×	−3	1	×	−2	
6	θ''	×		**0.231**	×		
7	Δ''	×	1	×	×	1	
1					1		
2				1			
3		1					
4	C'''		−1		1	−1	3
5	μ'''						
6	θ'''			×			
7	Δ'''						

We see that the program Y reduces only the first restraint in (7.34) ($r=1$) to an equality. We eliminate one variable, say y_3, from the system. The main part of the new tableau (1st tableau) is transformed according to recurrence formulas (7.26). The direction element of the transformation is $a_{31}=1$. We have, for example

$$a'_{21}=\frac{a_{21}}{a_{31}}=\frac{2}{1}=2,$$
$$a'_{20}=a_{20}-a'_{21}a_{30}=3-2\cdot 3=-3.$$

In our case $r=1$. Therefore, in the first step, we transform the row Δ also. The parameters Δ' are computed from

$$\Delta'_j=\Delta_j-\theta'_0\mu'_j. \tag{7.35}$$

In the $(m+2)$-th (fifth) row of the 1-st tableau we give the values of μ'_j. The parameters μ'_j are computed as products of columns $\overline{A}_0$ and $\overline{A}_j$ (the first m entries) in the 1st tableau. In taking the product, only the elements of nontransformed rows are taken into consideration. Thus, for example,

$$\mu'_2=a'_{12}a'_{10}+a'_{22}a'_{20}=0\,(-1)+(-3)\,(-3)=9.$$

In the $(m+3)$-th row of the 1st tableau we write $\theta'_j=\frac{\Delta_j}{\mu'_j}$. The least value θ'_0 of the row θ' is $\frac{1}{13}\cong 0.0769$; it is obtained on A_4. Applying θ'_0 and the parameters μ'_j, we compute from (7.35) the elements Δ'_j of the $(m+4)$-th row (Δ') in the 1st tableau. Now, single out the column $\overline{A}_4$ with $\Delta'_4=0$ and repeat the entire sequence of operations which leads to the 2nd tableau.

In the 2nd tableau three columns are singled out. Starting with the last of these columns, we fill the last, 3rd, tableau.

The components of the support program $Y=(y_1,\ y_2,\ y_3)$ are obtained from the $(m+1)$-th, 4th, row. To determine y_i we move along the i-th row of the 3rd tableau to the position in which 1 is written. The corresponding entry in the $(m+1)$-th, 4th, row is the component required. Thus,

$$y_1=1,\quad y_2=0,\quad y_3=0.$$

There is thus no need to fill the rows μ''', θ''', Δ'''.

It can easily be verified that the support program $Y=(1,\ 0,\ 0)$ solves problem (7.33)–(7.34).

The first support program obtained need not be the optimal program of the problem in question. In general, when the support program obtained by the gradient method does not solve the problem, the dual method should be further applied. To do so, besides the support program, the corresponding matrix $\|e_{ij}\|$ of the coefficients in the expansion of the m-dimensional vectors $e_j\ (j=1,\ 2,\ \ldots,\ m)$ in terms of the basis obtained is also required. The elements e_{ij} of this matrix satisfy the equations

$$\sum_{\lambda=1}^{m} a_{\lambda s_j}e_{i\lambda}=\begin{cases}1 \text{ for } i=j,\\ 0 \text{ for } i\neq j.\end{cases} \tag{7.36}$$

Here $I_Y=(s_1,\ \ldots,\ s_m)$ is the set of indices of the basis vectors.

The system of equations

$$\sum_{i=1}^{m} a_{ij}y_i=c_j,\quad j\in I_y,$$

defining the components y_i of the support program Y differs from system (7.36) only in the right-hand side. It is, therefore, advisable to solve these systems simultaneously. If we know in advance which of the restraint vectors enter the required basis, we could have applied the gradient algorithm to obtain, together with the support program, the corresponding matrix $\|e_{ij}\|$. To do so we would have to write the unit row vectors corresponding to the basis vectors in the rows $m+2,\ m+3,\ \ldots,\ 2m+1$ of the zeroeth tableau and successively transform them following the formulas used in transforming the main part of the gradient tableaus. As a result, simultaneously with the support program, we would have obtained the matrix $\|e_{ij}\|$

at the intersection of the additional rows and the basis columns. Note that gradient transformations do not affect the unit vector e_j until the components of the vector $\bar{A}_j$ have been inserted into the system of equalities (7.12). Therefore, the additional unit row vectors can be introduced gradually, as the new basis is being formed. This enables us to combine the construction of the matrix $\|e_{ij}\|$ with the determination of the support program Y.

We proceed from the last computational tableau of the gradient method (say the N-th tableau) in order to fill the main part of the principal zeroeth tableau when solving the problem under the second dual algorithm.

The element e_{ij} of the principal zeroeth tableau ($i=1, 2, \ldots, m$; $j=1, 2, \ldots, m$) is written in the N-th tableau in the row corresponding to the vector e_{s_i}. Here s_i is the index of the vector A_{s_i} occupying the i-th basis position. The components of the support program Y are written in the $(m+1)$-th row of the N-th tableau.

To obtain e_{ij} (the element in the j-th position in the i-th row of the principal zeroeth tableau) we must locate the unit in the j-th row ($j=1, 2, \ldots, m$) of the N-th tableau. The element of the row e_{s_i} in the N-th tableau located in the column containing this unit is the required e_{ij}. The element in the $(m+1)$-th row located in the same column specifies the component $y_j=e_{m+1, j}$ of the initial support program Y. The basis components of the initial pseudoprogram (elements of the column e_0 in the principal zeroeth tableau) are computed from the general formulas

$$x_{i0}=\sum_{\lambda=1}^{m} e_{i\lambda}b_\lambda .$$

We now illustrate the above.

Example 2. Using the dual method, maximize the linear form

$$L(X)=x_1+2x_2+3x_3+x_4+2x_5+3x_6+x_7 \tag{7.37}$$

subject to the conditions

$$\left.\begin{aligned} x_1+x_2+2x_3+3x_4+2x_5+3x_6+x_7&=7,\\ 2x_1+3x_2+x_3+x_4+3x_5+2x_6+2x_7&=8,\\ x_1+2x_2+3x_3+2x_4+0{,}5x_5+x_6+x_7&=6,\\ 2x_1+x_2+3x_3+x_4+2x_5+3x_6+x_7&=7,\end{aligned}\right\} \tag{7.38}$$

$$x_j\geqslant 0,\quad j=1, 2, \ldots, 7.$$

Solution. The corresponding dual problem is stated as follows:
Minimize the linear form

$$\tilde{L}(Y)=7y_1+8y_2+6y_3+7y_4$$

subject to the conditions

$$\begin{aligned} y_1+2y_2+y_3+2y_4&\geqslant 1,\\ y_1+3y_2+2y_3+y_4&\geqslant 2,\\ 2y_1+y_2+3y_3+3y_4&\geqslant 3,\\ 3y_1+y_2+2y_3+y_4&\geqslant 1,\\ 2y_1+3y_2+0.5y_3+2y_4&\geqslant 2,\\ 3y_1+2y_2+y_3+3y_4&\geqslant 3,\\ y_1+2y_2+y_3+y_4&\geqslant 1.\end{aligned}$$

All the components of the restraint vectors are positive. Therefore, following the suggestions of 7-2, we take as the initial feasible program any vector of the system

$$Y_1=(y_1, 0, 0, 0);\quad Y_2=(0, y_2, 0, 0);\quad Y_3=(0, 0, y_3, 0);$$
$$Y_4=(0, 0, 0, y_4),$$

where

$$y_i\geqslant \max_{1\leqslant j\leqslant 7}\frac{c_j}{a_{ij}}.$$

Starting with the feasible program $Y=Y_3=(0, 0, 4, 0)$, we compute the support program and the corresponding parameters e_{ij} by the gradient method.

TABLE 6.19 (0–4)

			↓	↓		↓	↓		
No.	A_0	$\bar{A}_1$	$\bar{A}_2$	$\bar{A}_3$	$\bar{A}_4$	$\bar{A}_5$	$\bar{A}_6$	$\bar{A}_7$	No. of tableau
1	7	1	1	2	3	**2**	3	1	0
2	8	2	3	1	1	3	2	2	
3	6	1	2	3	2	0,5	1	1	
4	7	2	1	3	1	2	3	1	
5	C	1	2	3	1	2	3	1	
6	e_5					1			
7	Δ	3	6	9	7	×	1	3	
1						1			1
2	−2.5	0.5	1.5	−2	−3.5	1.5	−2.5	0.5	
3	4.25	0.75	1.75	2.5	1.25	0.25	**0.25**	0.75	
4		1		1	−2	1			
5	C'		1	1	−2	1			
6	e_5	−0.5	−0.5	−1	−1.5	0.5	−1.5	−0.5	
7	e_6						1		
8	μ'	1.938	3.688	15.625	14.062	×	7.312	1.938	
9	θ'	1.548	1.627	0.576	0.498	×	**0.137**	1.548	
10	Δ'	2.735	5.496	6.863	5.077	×	×	2.735	

TABLE 6.19 (continued)

No.	A_0	$\bar{A}_1$	$\downarrow$ $\bar{A}_2$	$\downarrow$ $\bar{A}_3$	$\bar{A}_4$	$\downarrow$ $\bar{A}_5$	$\downarrow$ $\bar{A}_6$	$\bar{A}_7$	No. of tableau
1						1			2
2	40	8	**19**	23	9	4	−10	8	
3							1		
4		1		1	−2	1			
5	c''		1	1	−2	1			
6	e_5	4	10	14	6	2	−6	4	
7	e_6	−3	−7	−10	5	−1	4	−3	
8	e_2		1						
9	μ''	320	760	920	360	×	×	320	
10	θ''	0.00855	**0.00723**	0.00746	0.0141	×	×	0.00855	
11	Δ''	0.421	×	0.211	2.474	×	×	0.421	
1						1			3
2		1							
3							1		
4		1		**1**	−2	1			
5	G'''	−0.421	0.0526	−0.211	−2.474	0 789	0.526	−0.421	

TABLE 6.19 (continued)

			↓	↓		↓	↓		
No.	A_0	$\bar{A}_1$	$\bar{A}_2$	$\bar{A}_3$	$\bar{A}_4$	$\bar{A}_5$	$\bar{A}_6$	$\bar{A}_7$	No. of tableau
6	e_5	−0.211	0.526	1.895	1.263	−0.105	−0.737	−0.211	3
7	e_6	−0.0526	−0.368	−1.526	−1.684	0.474	0.316	−0.0526	
8	e_2	−0.421	0.0526	−1.211	−0.474	−0.211	0.526	−0.421	
9	e_3			1					
10	μ'''	1	×	1	−2	×	×		
11	θ'''	0.421	×	0.211		×	×		
12	Δ'''	0.211	×	×	2.895	×	×	0.421	
1						1			4
2			1						
3							1		
4				1					
5	C^{IV}	−0.211	0.0526	−0.211	−2.895	1	0.526	−0.421	
6	e_5	−2.105	0.526	1.895	5.053	−2	−0.737	−0.211	
7	e_6	1.474	−0.368	−1.526	−4.737	2	0.316	−0.0526	
8	e_2	0.789	0.0526	−1.211	−2.895	1	0.526	−0.421	
9	e_3	−1		1	2	−1			
10	μ^{IV}								
11	θ^{IV}				×				
12	Δ^{IV}								

The entire computational procedure is given in Table 6.19 (0–4). In the 1st tableau, the row Δ gives the differences between the left- and the right-hand sides of the restraints of the dual problem for the given feasible program:

$$\Delta_j = 4a_{3j} - c_j, \qquad j = 1, 2, \ldots, 7.$$

The row Δ has one zero corresponding to the vector A_6. In the zeroeth tableau we should, therefore, add a row whose elements are the components of the unit vector $e_5 = (0, 0, 0, 0, 1, 0, 0)$. The rows e_j are best arranged directly below the rows in the main part of the tableau, so as to facilitate their transformation by means of the same recurrence formulas.

The column $\overline{A}_5$ is taken as the direction column of the transformation. Any of the rows with a nonzero element in the position corresponding to column $\overline{A}_5$ can be taken as the direction row. To be specific, we choose the first row. The direction element of the transformation $a_{15} = 2$. The main part and the row e_5 of the zeroeth tableu are transformed according to recurrence formulas (7.26). The parameters μ', θ', and Δ'_j are then computed and written in the corresponding rows of the 1st tableau. Transformation to the 2nd tableau is carried out analogously.

When filling the 3rd tableau we encounter the special case mentioned in 7-3:

$$a'''_{i0} = b'''_i = 0, \qquad i = 1, 2, 3, 4.$$

According to the rule discussed in 7-4 (formula (7.28)), one of the nontransformed rows should be transferred to row μ'''. In our case, there remains a single nontransformed row, namely the fourth. Therefore $\mu'''_j = a'''_{4j}$.

Some of the elements in row μ''' are positive. In the positions of the row θ''' corresponding to $\mu''' > 0$ we write

$$\theta'''_j = \frac{\Delta''_j}{\mu'''_j}.$$

The positions of the row θ''' for which $\mu''' \leq 0$ are crossed out. The parameter θ'''_0 is determined from (7.29), and the row Δ''' is filled as usual.

Formation of the 4th tableau completes the gradient method. The 4th tableau contains all the information needed on the required support program $\overline{Y} = (\overline{y}_1, \ldots, \overline{y}_m)$ of the dual problem and the corresponding matrix $\|e_{ij}\|$. The basis of the program comprises the vectors A_2, A_3, A_5, and A_6.

In Table 6.20 (principal tableaus 0–1) and in Table 6.21 (auxiliary tableau) we show the process of solution of problem (7.37)–(7.38) according to the second dual algorithm. The principal zeroeth tableau has been filled according to the data obtained when computing the support program $\overline{Y}$ by the gradient method (see the last 4th tableau of Table 6.19). Thus, for instance, e_{33}, corresponding to the restraint vector $A_3 = A_{s_3}$, is taken from the 4th tableau (Table 6.19) from the row $e_{s_3} = e_5$. In the row $j=2$ of the 4th tableau the unit is in column $\overline{A}_2$. The parameter e_{32} is at the intersection of row e_5 and column $\overline{A}_2$: $e_{32} = 0.526$. The element $e_{14} = 2$ is located in the 4th tableau (Table 6.19) at the intersection of row $e_{s_4} = e_6$ and column A_5, which contains the only unit of the row $j = 1$.

The components of the initial support program Y, the elements in the last row of the principal zeroeth tableau, are determined in the same manner. In this case the $(m+1)$-th row of the 4th tableau replaces the rows e_j. Thus, for example, $y_1 = e_{m+1,1} = 1$, $y_2 = e_{m+1,2} = 0.0526$, etc.

The basis components x_{i0} of the initial pseudoprogram, i.e., the elements in column e_0 of the initial zeroeth tableau, are computed as products of the elements $e_{i\lambda}$ of the i-th row of the tableau and the corresponding components of the constraint vector. Thus, e.g.,

$$x_{30} = -2 \cdot 7 + 0.526 \cdot 8 - 0.737 \cdot 6 + 1.895 \cdot 7 = -0.947.$$

The problem is then solved by the general rules of the second dual algorithm.

The optimal program of the primal problem is

$$X^* = (0,\ 1.562,\ 0.375,\ 0.188,\ 0;\ 1.375,\ 0).$$

The solution of the dual problem is

$$Y^* = (-0.146,\ 0.354,\ 0.104,\ 0.875).$$

The optimal value of the linear form is

$$L(X^*) = \tilde{L}(Y^*) = 8.562.$$

One iteration of the dual method was required in this case to solve the problem. This is due to the fact that the gradient method yielded a good first approximation.

As we have already observed, this is not a chance phenomenon: the gradient method generally produces a support program which is close to the optimum.

TABLE 6.20

				B	7	8	6	7		No. of tableau
	No.	C	B	e_0	e_1	e_2	e_3	e_4	A_k	
	1	2	A_2	2.105	1	0.0526	0.526	—1.211	2.895	0
	2	3	A_3		—1			1	—2	
←	3	2	A_5	—0.947	—2	0.526	—0.737	1.895	5.053	
	4	3	A_6	2.263	2	—0.368	0.316	—1.526	4.737	
	5	—	L	9.105	1	0.0526	0.526	—0.211	2.895	
	1	2	A_2	1.562	—0.146	0.354	0.104	—0.125		1
	2	3	A_3	0.375	0.208	—0.208	0.292	0.25		
→	3	1	A_4	0.188	0.396	—0.104	0.146	—0.375		
	4	3	A_6	1.375	0.125	0.125	—0.375	0.25		
	5	—	L	8.562	—0.146	0.354	0.104	0.875		

TABLE 6.21

No.	B	A_1	A_2	A_3	A_4	A_5	A_6	A_7	Y_0	A_5	Y_1
1	7	1	1	2	3	2	3	1	1	—2	—0,146
2	8	2	3	1	1	3	2	2	0.0526	0.526	0.354
3	6	1	2	3	2	0.5	1	1	0.526	—0.737	0.104
4	7	2	1	3	1	2	3	1	-0.211	1.895	0.875
5	c_j	1	2	3	1	2	3	1			
0	Δ	0.211			2.895			0.421	$\theta_0=0.573$		
	$Z_{(5)}$	2.105			—5.053	1		0.211			
	$-\Delta/Z_{(5)}$				**0.573**						
1	Δ'	1.417				0.573		0.542			

7-6. When determining the initial support program by the gradient method we assumed that some feasible program of the problem was known. We mentioned before that for a wide class of problems the computation of a feasible program involves no difficulties. There are, nevertheless, problems in which the operations needed to find an initial feasible program are no less tedious than those necessary to compute a support program from a feasible program. In these cases it is advisable to follow the method discussed below.

To restraints (1.2) of the linear-programming we add the condition

$$x_1 + x_2 + \ldots + x_n \leqslant M,$$

or

$$x_0 + x_1 + \ldots + x_n = M, \qquad x_0 \geqslant 0.$$

The new problem (in variables $x_0, x_1, \ldots, x_n$) is called the a u g m e n t e d p r o b l e m.

The dual problem involves minimization of the linear form

$$My_0 + \sum_{i=1}^{m} b_i y_i \tag{7.39}$$

subject to the conditions

$$y_0 + \sum_{i=1}^{m} a_{ij} y_i \geqslant c_j, \qquad j = 1, 2, \ldots, n,$$

$$y_0 \geqslant 0.$$

This is the a u g m e n t e d d u a l p r o b l e m.

The vector

$$Y = (y_0, 0, \ldots, 0),$$

where

$$y_0 = \max_j \{c_j, 0\}.$$

can, obviously, be taken as a feasible program of the augmented dual problem.

Starting with program Y, we may apply the gradient method to derive a support program of the augmented dual problem. The resulting dual basis will be taken as the basis of the initial pseudoprogram and the augmented problem is then solved by the dual method. In the process of solution, M is assumed fairly large (larger than any number with which it is to be compared in the process).

Let $\overline{X}^* = (x_0^*, x_1^*, \ldots, x_n^*)$ and $\overline{Y}^* = (y_0^*, y_1^*, \ldots, y_m^*)$ be the optimal programs of the primal and the dual augmented problems. Two cases may arise

$$y_0^* = 0 \text{ or } y_0^* > 0.$$

In the first case $Y^* = (y_1^*, \ldots, y_m^*)$ is obviously a solution of the dual problem (1.4)–(1.5). Therefore $X^* = (x_1^*, \ldots, x_n^*)$ is an optimal program of problem (1.1)–(1.3):

$$\sum_{j=1}^{n} c_j x_j^* + 0x_0^* = \sum_{i=1}^{m} b_i y_i^* + M \cdot 0.$$

In the second case ($y_0^* > 0$), the set of feasible programs of the dual problem is empty. Indeed, let $Y = (y_1, \ldots, y_m)$ be a feasible program of problem (1.4)–(1.5). Then $\overline{Y} = (0, y_1, \ldots, y_m)$ is a feasible program of the augmented problem. For sufficiently large M, the value of the linear form (7.39) at

the point $Y^* = (y_0^*, y_1^*, \ldots, y_m^*)$ will be higher than its value at the point $\bar{Y} = (0, y_1, \ldots, y_m)$, and this contradicts the optimality of program Y^*. Thus, for $y_0^* > 0$, problem (1.4)–(1.5), and, therefore, its dual problem (1.1)–(1.3) are unsolvable.

If when solving the augmented problem we obtain case (b), indicating inconsistency of restraints, this is a hint that the restraints of the primal problem are also inconsistent. Indeed, if $X = (x_1, \ldots, x_n)$ is a feasible program of the primal problem then for $M \geqslant \sum_{j=1}^{n} x_j$ the vector X is a feasible program of the augmented problem also.

Thus, this method enables us to find an optimal program of a solvable problem without first determining a feasible program of the dual problem.

7-7. In conclusion we give another technique for solving linear-programming problems by the dual method, without preliminary tedious computations of an initial dual basis. The technique is based on the proposition (3-7) that in a linear-programming problem with bilateral restraints any system of m linearly independent restraint vectors can be taken as the basis of an initial pseudoprogram.

We take M (a number larger than any comparand to be used in the process of solution) as the upper bound of some variables in problem (1.1)–(1.3). The variables to be bounded above will be specified in what follows. The new problem will be called the bounded problem.

Physical meaning or some features of the matrix of coefficients may enable us to isolate m linearly independent restraint vectors. This system can then be considered as the basis of some pseudoprogram of the problem.

In the general case, when the features of the problem do not hint at a linearly independent system of vectors, the dual basis can be determined by successive transformation of equalities (1.2), following the complete reduction method. The components of the restraint vectors are transformed according to recurrence formulas of the form (5.16):

$$a_{ij}^{(l+1)} = \begin{cases} a_{ij}^{(l)} - \dfrac{a_{rj}^{(l)}}{a_{rk}^{(l)}} a_{ik}^{(l)}, & i \neq r; \quad j = 1, 2, \ldots, n. \\ \dfrac{a_{rj}^{(l)}}{a_{rk}^{(l)}}, \quad i = r, & \end{cases}$$

As the direction element in each successive step we take any nonzero element which does not belong to the direction rows or columns of the preceding steps. It may, however, turn out that at a certain step this rule leaves us with no direction elements.

The direction rows and columns of the preceding steps are crossed out in the matrix. The resulting array is called a *reduced matrix*. The transformation of the restraint matrix and of the constraint vector should be discontinued when the reduced matrix becomes empty or a null matrix. The reduced matrix may become empty at some step if restraints (1.2) are consistent and linearly independent. In this case, the restraint vectors transformed in the preceding steps into unit vectors can be taken as the basis of the initial pseudoprogram. A reduced matrix is null in one of the two cases:

(i) the problem restraints are inconsistent; in this case at least one of the components $b_i^{(l)}$ of the transformed restraint vector corresponding to a row of the reduced matrix is nonzero;

(ii) the problem restraints corresponding to the rows of the reduced matrix are linearly dependent on the other restraints; in this case $b_i^{(l)}=0$ for all rows of the reduced matrix; the rows isolated in this way can be crossed out; the rank of the restraint matrix is now lower.

These transformations of the matrix $\|a_{ij}\|$ will, thus, eventually establish the unsolvability of the problem or produce a system of linearly independent restraint vectors, i. e., the basis of the initial pseudoprogram. In the course of transformation, we also obtain the coefficients $x_{ij}^{(l)}$ in the expansion of the restraint vectors in terms of the vectors of the initial basis. These parameters are sufficient for solving the problem by the first dual algorithm.

The second algorithm also requires the inverse matrix

$$\|e_{ij}\|_m = \|a_{is_j}\|_m^{-1}$$

$(s_1, s_2, \ldots, s_m) = I_y$. The matrix $\|e_{ij}\|_m$ can be obtained together with the determination of the basis vectors if, from the start, we introduce to the right of the restraint matrix m unit (m-dimensional) vectors e_j and transform the augmented matrix according to the above rules.

A sequence of transformations will thus produce the complete system of linearly independent restraint vectors $A_{s_1}, \ldots A_{s_m}$ and the corresponding matrix $\|x_{ij}\|_{m,n}$ (or $\|e_{ij}\|_m$ of the expansion coefficients of the vectors A_j (or e_j) in the linearly independent system chosen. We then compute the parameters

$$\Delta_j = \sum_{i=1}^{m} c_{s_i} x_{ij} - c_j.$$

The variables x_j whose Δ_j are negative are bounded above by a sufficiently large number M. We have thus obtained the b o u n d e d p r o b l e m. The system of restraint vectors $(A_{s_1}, \ldots, A_{s_m})$ is the basis of a pseudoprogram of this problem.

We should emphasize that in this bounded problem the variables with $\Delta_j \geqslant 0$ (the basis components of the pseudoprogram included) are restrained only by the requirement of nonnegativity, whereas the remaining variables with negative Δ_j are bilaterally restrained.

The bounded problem is solved by the dual method. The remarks in 5-5 and 6-5 concerning problems with bilateral restraints are taken into consideration. In the process of solution the number M is taken larger than any comparand which need arise.

It follows from the preceding that in a finite number of steps we obtain case (a) or case (b).

In case (a), the optimal value of the linear form of the bounded problem, as in any problem with bilateral restraints, is written according to (3.11) in the form

$$L(X) = \sum_{i=1}^{m} b_i y_i - \sum_{j \notin I_y} \Delta_j \gamma_j.$$

Here y_i are the components of the kernel Y of the solution of the problem dual with respect to the bounded problem, I_y is the set of indices of the basis vectors of the vectors of the optimal program X, γ_j is the value of the extrabasis components of the solution. In our case

$$\gamma_j = \begin{cases} M & \text{if } j \in E, \\ 0 & \text{if } j \notin E, \end{cases}$$

where E is the set of indices of the extrabasis variables of the solution of the bounded problem equal to the upper bound M.

Thus, in case (a) the maximum value of the linear form of the bounded problem can always be written as

$$L(X)=L_1(X)+ML_2(X)=L_1(X)-M\sum_{j\in E}\Delta_j. \tag{7.40}$$

Here

$$L_2(X)=-\sum_{j\in E}\Delta_j\geqslant 0. \tag{7.41}$$

The basis components x_{i_0} of the solution X of the bounded problem are the coefficients in the expansion of the vector $B-\sum_{j\in Y}A_j\gamma_j$ in terms of the basis vectors of the optimal program X,

$$x_{i_0}=b_i^{(Y)}-\sum_{j\in Y}x_{ij}^{(Y)}\gamma_j,$$

or, equivalently,

$$x_{i_0}=b_i^{(Y)}-M\sum_{j\in E}x_{ij}^{(Y)}. \tag{7.42}$$

In case (a) two outcomes are possible:

$$\text{(i) } L_2(X)=0, \quad \text{(ii) } L_2(X)\neq 0.$$

In (i) the linear form is independent of M. The primal problem is solvable and its optimal program can easily be determined. Case (a) arises when the set E is empty or when $\sum_{j\in E}\Delta_j=0$. If the set E is empty, all the extrabasis variables of program X are zero and the solution of the primal problem coincides with the optimal program X of the bounded problem.

The basis components of the solution $x_{i_0}=b_i^{(Y)}$ (see (7.42)) and are independent of M. If $\sum_{j\in E}\Delta_j=0$, then from (7.41) $\Delta_j=0$ for $j\in E$ (the optimal program of the dual problem is degenerate). For the other (zero) extrabasis variables

$$\Delta_j\geqslant 0.$$

According to the optimality criterion (see Chapter 3, Theorem 5.2) a feasible program satisfying these conditions is an optimal program of the primal problem, though not necessarily a support program. As the parameter M entering (7.42), we may take any $M_0\geqslant 0$ for which

$$x_{i_0}=b_i^{(Y)}-M_0\sum_{j\in E}x_{ij}^{(Y)}\geqslant 0.$$

A support program can be obtained from the feasible program (if at all necessary) without difficulty. The computations are carried out according to the gradient method.

In case (ii), when

$$L_2(X)=-\sum_{j\in E}\Delta_j>0,$$

the primal problem is unsolvable, since the linear form is unbounded in the set of feasible programs.

In case (b), when the bounded problem is unsolvable, the primal problem

is not solvable either. The unsolvability of the problem can be either due to inconsistency of restraints or to the fact that the linear form is unbounded above.

A rigorous theory of this method is based on the following two propositions:

1. There exists a number $M_1<\infty$ such that for all $M>M_1$ the linear form of problem (1.1)–(1.3) is unbounded above in the set of its feasible programs when $\sum_{j\in E}\Delta_j>0$

2. There exists a number $M_2<\infty$ such that for all $M>M_2$ unsolvability of the bounded problem implies unsolvability of the primal problem (1.1)–(1.3).

The proof of these propositions is analogous to the proof of the corresponding theorems in the M-method. The reader, following the technique of Chapter 4, § 7, will easily establish the validity of these propositions (see Exercise 9).

We illustrate the preceding by an appropriate example.

Example. Maximize the linear form

$$L(X)=10x_1+2x_2+x_3+9x_4+2x_5+x_6+x_7$$

subject to the conditions

$$\begin{aligned}
3x_1+7x_2+5x_3+4x_4+x_5+x_6 &=5,\\
7x_1+15x_2+9x_3+9x_4+x_5+2x_6+x_7&=8,\\
6x_1+11x_2+7x_3+6x_4+x_5+x_6+x_7&=6,\\
x_j\geq 0,\ j=1,\ 2,\ \dots,\ 7.
\end{aligned}$$

Solution. We must first obtain a system of linearly independent vectors. In Table 6.22 (0–3) we give the sequence of transformations leading to the linearly independent vectors A_5, A_6, and A_7 and the coefficients in the expansion of all the restraint vectors in terms of these linearly independent vectors. The primal problem is then (after the 3rd step) replaced by the following problem.

TABLE 6.22 (0–3)

No.	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	No. of tableau
1	5	3	7	5	4	1	1		0
2	8	7	15	9	9	1	2	1	
3	6	6	11	7	6	1	1	1	
1	5	3	7	5	4	1	1		1
2	3	4	8	4	5		1	1	
3	1	3	4	2	2			1	
1	2	−1	−1	1	−1	1		−1	2
2	3	4	8	4	5		1	1	
3	1	3	4	2	2			1	
1	3	2	3	3	1	1			3
2	2	1	4	2	3		1		
3	1	3	4	2	2			1	

Maximize the linear form

$$L(X)=10x_1+2x_2+x_3+9x_4+2x_5+x_6+x_7 \tag{7.43}$$

TABLE 6.23 (0-3)

					C	10	2	1	9	2	1	1			
					$(\alpha;\beta)$	0; M	0; ∞	0; ∞	0; M	0; ∞	0; ∞	0; ∞			No. of tableau
	No.	C_X	B_X	A_0	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	δ	$(\alpha;\beta)_x$	
	1	2	A_5	$3-3M$	$3-M$	2	3	3	1	1			$3M-3(\alpha)$	0; ∞	0
	2	1	A_6	$2-4M$	$2-3M$	1	4	2	3		1		$4M-2(\alpha)$	0; ∞	
←	3	1	A_7	$1-5M$	$1-2M$	3	4	2	2			1	$5M-1(\alpha)$	0; ∞	
	4	—	Δ	$9+4M$	—	-2	12	9	-2				—	—	
	5	—	X	—	—	β	α	α	β				—	—	
	6	—	θ	—	—	2/3	—	—	1				—	—	
	1	2	A_5	$7/3+1/3M$	$7/3$		$1/3$	$5/3$	$-1/3$	1		$-2/3$	—	0; ∞	1
←	2	1	A_6	$5/3-7/3M$	$5/3$		$8/3$	$4/3$	$7/3$		1	$-1/3$	$7/3M-5/3(\alpha)$	0; ∞	
→	3	10	A_1	$1/3-2/3M$	$1/3$	1	$4/3$	$2/3$	$2/3$			$1/3$	$2/3M-1/3(\alpha)$	0; M	
	4	—	Δ	$29/3+2/3M$	—		$44/3$	$31/3$	$-2/3$			$2/3$	—	—	
	5	—	X	—	—		α	α	β			α	—	—	
	6	—	θ	—	—		—	—	2/7			2	—	—	

TABLE 6.23 (continued)

					C \ $(\alpha;\beta)$	10	2	1	9	2	1	1			
						0; M	0; ∞	0; ∞	0; M	0; ∞	0; ∞	0; ∞			No. of tableau
	No.	C_X	B_X	A_0	$\tilde{A}_0$	A_1	A_2	A_3	A_4	A_5	A_6	A_7	δ	$(\alpha;\beta)_x$	
	1	2	A_5	$^{18}/_7$			$^5/_7$	$^{13}/_7$		1	$^1/_7$	$-^5/_7$			2
⟶	2	9	A_4	$^5/_7$			$^8/_7$	$^4/_7$	1		$^3/_7$	$-^1/_7$			2
⟵	3	10	A_1	$-^1/_7$		1	$^4/_7$	$^2/_7$			$-^2/_7$	$^3/_7$			2
	4	—	Δ	$^{71}/_7$			$^{108}/_7$	$^{75}/_7$			$^2/_7$	$^4/_7$			2
	5	—	X	—											2
	6	—	θ	—			—	—			**1**	—			2
	1	2	A_5	$^5/_2$		$^1/_2$	1	2		1		$-^1/_2$			3
	2	9	A_4	$^1/_2$		$^3/_2$	2	1	1			$^1/_2$			3
⟶	3	1	A_6	$^1/_2$		$-^7/_2$	-2	-1			1	$-^3/_2$			3
	4	—	Δ	10		1	16	11				1			3
	5	—	X	—											3
	6	—	θ	—											3

subject to the conditions

$$\left.\begin{array}{l} 2x_1+3x_2+3x_3+\ x_4+x_5 \qquad\quad =3, \\ x_1+4x_2+2x_3+3x_4 \qquad +x_6 \qquad =2, \\ 3x_1+4x_2+2x_3+2x_4 \qquad\qquad +x_7=1, \end{array}\right\} \tag{7.44}$$

$$x_j \geqslant 0, \ j=1, 2, \ldots, 7. \tag{7.45}$$

The presence of unit restraint vectors in problem (7.43)–(7.45) and of positive right-hand sides in system (7.44) indicates that it is advisable to apply the simplex method. Nevertheless, for the sake of illustration, we employ the first dual algorithm.

We compute the parameters Δ_j corresponding to the restraint vectors. We have

$$\Delta_1=-2, \quad \Delta_2=12, \quad \Delta_3=9, \quad \Delta_4=-2.$$

Following the suggestions of the present article, we take M as the upper bound of the variables with negative Δ_j. The bounded problem corresponding to problem (7.43)–(7.45) requires maximization of linear form (7.43) subject to conditions (7.44), (7.45) and the additional restraints

$$x_1 \leqslant M, \quad x_4 \leqslant M. \tag{7.46}$$

In Table 6.23 (0–3) we give the process of solution of the bounded problem by the first dual algorithm. The computations are carried out as suggested in 5–5, where the first algorithm is applied to problems with bilateral restraints.

We see from the tableaus that after the first iteration M is omitted from the expressions for pseudo-program components and linear-form coefficients. The computations in subsequent iterations can be carried out effectively according to the first algorithm for the problem in canonical form. There is, thus, no need to fill the row X and the columns $\bar{A}_0$, δ, and $(\alpha;\ \beta)_X$.

After three iterations, we obtain the solution

$$X=(0, 0, 0, 1/2, 5/2, 1/2, 0).$$

§ 8. The simplex method and the dual method

8-1. In the present chapter the dual method was presented independently of the simplex method. We shall now show that the dual method is actually a variation of the simplex method applied to the solution of the dual problem.

We rewrite the dual problem (1.4)–(1.5) in the following form:

Minimize the linear form

$$\tilde{L}(Y)=\sum_{i=1}^{m} b_i y_i \tag{8.1}$$

subject to the conditions

$$\sum_{i=1}^{m} a_{ij} y_i - y_{m+j} = c_j, \tag{8.2}$$

$$y_{m+j} \geqslant 0,$$

$$j=1, 2, \ldots, n. \tag{8.3}$$

Consider the vectors

$$A^{(i)}=(a_{i1}, a_{i2}, \ldots, a_{in})^T,$$

$$\bar{e}_j=(\underbrace{\overbrace{0, \ldots, 0, 1}^{j}, 0, \ldots, 0}_{n})^T,$$

$$C=(c_1, c_2, \ldots, c_n)^T,$$

and rewrite restraints (8.2) in the vector form

$$\sum_{i=1}^{m} A^{(i)} y_i - \sum_{j=1}^{n} \bar{e}_j y_{m+j} = C. \tag{8.4}$$

The primal problem (1.1)–(1.3) will be called problem (A) and the dual problem written in the form (8.1)–(8.3) problem $(\tilde{A})$. We shall solve problem $(\tilde{A})$ by the simplex method.

Problem $(\tilde{A})$ has $m+n$ variables $y_1, y_2, \ldots, y_{m+n}$ and n equality restraints. The restraint vectors $A^{(i)}$ corresponding to linearly independent (by assumption) equality restraints of the primal problem (A) are linearly independent. The variables $y_1, y_2, \ldots, y_m$ are not restrained by the requirement of nonnegativity. The system $A^{(1)}, A^{(2)}, \ldots, A^{(m)}$ therefore appears in the basis of any support program of problem $(\tilde{A})$. The basis of any support program $Y=(y_1, \ldots, y_m, y_{m+1}, \ldots, y_n)$ of this problem comprises n vectors (as the number of restraints (8.4)) and, besides the vectors $A^{(i)}$, it contains $n-m$ unit vectors $-e_j$. Let E_Y be the set of indices of the unit vectors appearing in the basis in question, and I_Y the set of indices of the remaining unit vectors. Thus,

$$I_Y=(s_1, s_2, \ldots, s_m).$$
$$E_Y=(s_{m+1}, s_{m+2}, \ldots, s_n),$$
$$E_Y+I_Y=\{1, 2, \ldots, n\}.$$

Consider the restraint vectors A_j of problem (A) for $j \in I_Y$.

The vectors $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$ constitute a linearly independent system. Indeed, the determinant of the corresponding matrix $A=(A_{s_1}, \ldots, A_{s_m})$ coincides, up to the sign, with the determinant of the matrix $\tilde{A}=(A^{(1)}, \ldots, A^{(m)}, -e_{s_{m+1}}, \ldots, -e_{s_n})$ of the restraint vectors of the support program Y of problem $(\tilde{A})$. This can easily be verified, if the determinant of the matrix $\tilde{A}$ is expanded in the elements of the columns corresponding to the vectors $-e_j (j \in E_Y)$.

In restraints (1.2) of problem (A) we take $x_j=0$ for $j \in E_Y$. Then, in view of the linear independence of vectors A_j $(j \in I_Y)$ restraints (1.2) define the set of numbers $x_j (j \in I_Y)$.

Let

$$x_{s_i}=x_{i0}, \quad i=1, 2, \ldots, m.$$

We denote by x_{ij}, $i=1, 2, \ldots, m$, $j=1,2, \ldots, n$, the corresponding coefficients in the expansion of the restraint vectors A_j in terms of the vectors of the linearly independent system $A_{s_1}, \ldots, A_{s_m}$. The following formulas can easily be verified (see, e.g., (1.15)):

$$x_{i0}=\sum_{\mu=1}^{m} b_\mu e_{i\mu} \tag{8.5}$$

$$x_{ij}=\sum_{\mu=1}^{m} a_{\mu j} e_{i\mu}, \tag{8.6}$$

$$i=1, 2, \ldots, m; \quad j=1, 2, \ldots, n.$$

Here

$$\| e_{i\mu} \| = \| a_{is_\mu} \|^{-1} \quad (i, \mu=1, 2, \ldots, m).$$

Thus, to each basis of problem $(\tilde{A})$ there correspond m linearly independent vectors A_{s_i}, $i=1, 2, \ldots, m$, and sets of numbers x_{i0} and x_{ij} satisfying relationships (8.5), (8.6).

We now compute the coefficients in the expansion of the restraint vectors and the constraint vector of problem $(\tilde{A})$ in terms of the basis vectors of support program Y. The restraint vectors of problem $(\tilde{A})$ not appearing

in the basis are the vectors e_j with $j \in I_Y$. The constraint vector of problem $(\tilde{A})$ is the vector C whose components are the linear-form coefficients of problem (A).

Let $Y_i = (y_{i1}, \ldots, y_{in})$ be the coefficients in the expansion of the vector $-e_{s_i}$ $(i = 1, \ldots, m)$ in terms of the basis vectors, and $Y_0 = (y_{01}, \ldots, y_{0n})$ the basis components of the support program Y of problem $(\tilde{A})$— the coefficients in the expansion of vector C in terms of the basis vectors.

By definition,

$$-e_{s_i} = \bar{A} Y_i, \quad i = 1, 2, \ldots, m, \tag{8.7}$$

$$C = \bar{A} Y_0, \tag{8.8}$$

where

$$\bar{A} = (A^{(1)}, \ldots, A^{(m)}, -e_{s_{m+1}}, \ldots -e_{s_n}).$$

For the sake of convenience we interchange the rows of matrix $\bar{A}$ and write

$$\bar{A} = \begin{Vmatrix} a_{1s_1} & \ldots & a_{ms_1} & 0 & \ldots & 0 \\ a_{1s_2} & \ldots & a_{ms_2} & 0 & \ldots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{1s_m} & \ldots & a_{ms_m} & 0 & \ldots & 0 \\ a_{1s_{m+1}} & \ldots & a_{ms_{m+1}} & -1 & \ldots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{1s_n} & \ldots & a_{ms_n} & 0 & \ldots & -1 \end{Vmatrix}.$$

From (8.7) and (8.8) we obtain

$$\begin{aligned} Y_i &= -\bar{A}^{-1} e_{s_i}, \quad i = 1, 2, \ldots m, \\ Y_0 &= \bar{A}^{-1} C. \end{aligned} \tag{8.9}$$

It is only natural to assume that the components of the vectors e_{s_i}, $i = 1$ $2, \ldots, m$, and of C in (8.9) and (8.10) are arrayed in the same way as the rows of the matrix $\bar{A}$.

Direct matrix multiplication will readily show that

$$\bar{A}^{-1} = \begin{Vmatrix} e_{11} & \ldots & e_{m1} & 0 & \ldots & 0 \\ e_{12} & \ldots & e_{m2} & 0 & \ldots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ e_{1m} & \ldots & e_{mm} & 0 & \ldots & 0 \\ e_{1,\,m+1} & \ldots & e_{m,\,m+1} & -1 & \ldots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ e_{1n} & \ldots & e_{mn} & 0 & \ldots & -1 \end{Vmatrix}. \tag{8.10}$$

Here

$$\| e_{i\mu} \| = \| a_{is_\mu} \|^{-1}, \quad i = 1, 2, \ldots, m; \quad \mu = 1, 2, \ldots, m;$$

$$e_{i\mu} = \sum_{\lambda=1}^{m} a_{\lambda s_\mu} e_{i\lambda}, \; i = 1, 2, \ldots, m, \mu = m+1, m+2, \ldots, n. \tag{8.11}$$

We rewrite relationships (8.9) and (8.10) using the explicit expression for the matrix $\bar{A}^{-1}$. We have

$$y_{ij} = -e_{ij}, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n; \tag{8.12}$$

$$y_{0j}=\begin{cases}\sum_{\lambda=1}^{m} c_{s_\lambda}e_{\lambda j} & \text{for } j=1,\ 2,\ \ldots,\ m;\\ \sum_{\lambda=1}^{m} c_{s_\lambda}e_{\lambda j}-c_{s_j} & \text{for } j=m+1,\ m+2,\ \ldots,\ n.\end{cases} \tag{8.13}$$

Comparing (8.6) and (8.11) we obtain

$$e_{i\mu}=x_{is_\mu},\quad i=1,\ 2,\ \ldots,\ m,\quad \mu=m+1,\ m+2,\ \ldots,\ n.$$

Hence, applying (8.12) and (8.13), we obtain

$$y_{i\mu}=-x_{is_\mu},\quad i=1,\ 2,\ \ldots,\ m,\quad \mu=m+1,\ \ldots,\ n, \tag{8.14}$$

$$y_{0\mu}=\sum_{\lambda=1}^{m} c_{s_\lambda}x_{\lambda s_\mu}-c_{s_\mu},\quad \mu=m+1,\ m+2,\ \ldots,\ n. \tag{8.15}$$

We shall now find relations for the relative evaluations of the restraint vectors of problems (A) and $(\tilde{A})$. To distinguish between the evaluations of the restraint vectors of these problems, we shall modify Δ with superscripts X and Y, respectively.

The evaluations of restraint vectors are expressed in terms of the corresponding linear-form coefficients and the coefficients in the expansion of these vectors in terms of the basis. According to the general rule of computation of the relative evaluations of restraint vectors, we have

$$\Delta_j^{(X)}=\sum_{\mu=1}^{m} x_{\mu j}c_{s_\mu}-c_j,\quad j\in E_Y; \tag{8.16}$$

$$\Delta_{s_i}^{(Y)}=\sum_{\mu=1}^{m} y_{i\mu}\,b_\mu+\sum_{\mu=1}^{m} y_{i,\,m+\mu}b_{m+s_\mu}-b_{m+s_i},\quad i=1,\ 2,\ \ldots,\ m.$$

By assumption, however, in problem $(\tilde{A})$, $b_j=0$ for $j>m$. Therefore, denoting $\Delta_{s_i}^{(Y)}$ by $\Delta_i^{(Y)}$, we obtain

$$\Delta_i^{(Y)}=\sum_{\mu=1}^{m} y_{i\mu}b_\mu,\qquad i=1,\ 2,\ \ldots,\ m.$$

Further, applying relationships (8.12) and (8.5), we obtain

$$\Delta_i^{(Y)}=-x_{i0},\qquad i=1,\ 2,\ \ldots,\ m. \tag{8.17}$$

From (8.15) and (8.16) we obtain

$$\Delta_{s_\lambda}^{(X)}=y_{0\lambda},\qquad \lambda=m+1,\ \ldots,\ n. \tag{8.18}$$

Formulas (8.17), (8.18), and (8.14) enable us to establish the complete equivalence of all the stages in the solution of problem $(\tilde{A})$ by the simplex method and of problem (A) by the dual method.

The parameters $y_{0\lambda}$ in (8.18) are the basis components of a support program of problem (A) restrained by the requirement of nonnegativity. Therefore, according to (8.18), $\Delta_j^{(X)}\geqslant 0$ for $j\in E_Y$ or $j\notin I_Y$. This indicates that the restraint vectors A_j, $j\in I_Y$ constitute the basis of a pseudoprogram of problem (A) with the basis components x_{i0}, $i=1,\ 2,\ \ldots,\ m$.

Thus, to each basis of the support program Y of problem $(\tilde{A})$ there corresponds a basis of the pseudoprogram of problem (A). The indices of the basis vectors and of the basis components of the pseudoprogram constitute the set I_Y defined by the basis of the support program Y of problem $(\tilde{A})$. The transition from one support program of problem $(\tilde{A})$ to another corresponds to a transition from one pseudoprogram of problem (A) to a successive one. The introduction of the vector $-e_j(j\in I_Y)$ into the vector of the

support program of problem $(\tilde{A})$ corresponds to elimination of the vector A_j from the basis of pseudoprogram of problem (A).

In problem $(\tilde{A})$ we minimize the linear form $\tilde{L}(Y)$. Hence, the optimality test of the solution of problem $(\tilde{A})$ has the form

$$\Delta_j^{(Y)} \leqslant 0; \qquad j \in I_Y,$$

or, equivalently from (8.17),

$$x_{i0} \geqslant 0; \qquad i=1,\ 2,\ \ldots,\ m.$$

The last relationships constitute the optimality test of the pseudoprogram of problem (A) corresponding to program Y of problem $(\tilde{A})$.

Case (a) of the simplex method, when applied to problem $(\tilde{A})$, is thus equivalent to case (a) of the dual method, when applied to problem (A).

Now let the support program Y of problem $(\tilde{A})$ not satisfy the optimality test. In this case some of the parameters $\Delta_i^{(Y)}$ are positive. Let $\Delta_r^{(Y)} > 0$. The condition for unsolvability of problem $(\tilde{A})$ is: for some $\Delta_r^{(Y)} > 0$ all $y_{r\mu} < 0$, $\mu = m+1,\ \ldots,\ n$. Formulas (8.17) and (8.14) show that with reference to problem (A) the requirement of unsolvability is stated as follows. To some negative basis component x_{r0} of the pseudoprogram there correspond nonnegative coefficients x_{rj} for all $j \in E_Y$.

Hence, case (b) of the simplex method for problem $(\tilde{A})$ coincides with case (b) of the dual method for problem (A).

Finally, let $\Delta_r^{(Y)} > 0$ and the unsolvability requirements not be satisfied (case (c)). In terms of problem $(\tilde{A})$ this indicates that the pseudoprogram of problem (A) defined by program Y of problem $(\tilde{A})$ has negative components and to each of these correspond negative x_{ij}. In other words, we have case (c) of the dual method.

Let us now analyze a single iteration in the solution of problem $(\tilde{A})$ by the simplex method. We introduce into the basis some vector $-e_{s_r}$ for which $\Delta_r^{(Y)} > 0$. The vector A_{s_r} is then eliminated from the basis of pseudoprogram of problem (A). Formula (8.17) shows that the vector A_{s_r} eliminated from the pseudoprogram basis corresponds to a negative basis component x_{r0}.

According to the simplex method, the vector $-e_{s_t}$ $(s_t \in E_Y)$ on which

$$\theta_0 = \min_{\substack{y_{ri} > 0 \\ i > m}} \frac{y_{0i}}{y_{ri}} = \frac{y_{0t}}{y_{rt}}$$

is obtained is eliminated from the basis of the support program of problem $(\tilde{A})$.

According to (8.18) and (8.14) the preceding condition indicates that the vector $A_k = A_{s_t}$ on which

$$\theta_0 = \min_{x_{rj} < 0} \left(-\frac{\Delta_j^{(X)}}{x_{rj}} \right) = -\frac{\Delta_k^{(X)}}{x_{rk}}$$

is obtained is introduced into the pseudoprogram basis of problem (A). We have thus established complete equivalence between the application of the simplex method to the dual problem and of the dual method to the primal problem.

8-2. When discussing the two algorithms of the dual method, we compared the bulkiness of computations in a single iteration of this method with the bulkiness of those in one iteration in the simplex method. We saw

that the computational procedure of the two methods are close in form, and the number of operations entailed in a single iteration is of the same order in the two methods. We remarked also that in general we have no information which would enable us to compare the number of iterations required for solving a problem by either of the two methods.

Taking into account the specific features of the two methods discussed in Chapters 4 to 6 we can choose one of the two in each concrete case, depending on the problem (or the class of problems) in question.

When choosing a method, considerable attention should be given to the determination of the initial support program. If, for example, the initial support program of the dual problem is obvious, whereas the initial support program of the primal problem is to be determined following the general rules, the problem should, obviously, be solved by the dual method.

In the following we shall discuss another feature of the dual method suitable for solving a wide class of problems.

The necessity often arises of solving several linear-programming problems differing in one or more restraints.

The exact statement of complicated applied problems studied by linear-programming methods generally leads to the solution of an entire series of problems differing in just a few restraints. The point is that when discussing new problems without any previous experience as to how they should be attacked, we are often unable to establish from the start all the restraints to be satisfied by the solution. An analysis of a tentative solution of the problem points to additional restraints which should be introduced. This process is often repeated more than once.

Similar difficulties arise when solving classical linear-programming problems. For example, when tackling a transportation problem we often disregard, in the first stages, the limited carrying capacity of the communication routes. Moreover, the carrying capacities of the communication routes often change in time. It is natural to require that program adjustment involve less operations than the initial determination of the transportation network.

The necessity in solving a series of linear-programming problems differing in a single restraint arises when the integer programming method is applied.

In all these cases it is, obviously, inadvisable to start solving the problem anew when some new restraints have been introduced. We must learn to utilize the information stored in the optimal program of the initial problem effectively. The dual method can be successively employed to this end.

Let the solution of an initial problem (we shall again call it problem (A)) yield the optimal support program $X^* = (x_1^*, \ldots, x_n^*)$ with the basis $A_{s_1}, \ldots, A_{s_m}$ We now have a new problem differing from problem (A) in the additional restraint

$$\sum_{j=1}^{n} a_{m+1,\, j} x_j \leqslant b_{m+1}, \tag{8.19}$$

or, equivalently,

$$\sum_{j=1}^{n} a_{m+1,\, j}\, x_j + x_{n+1} = b_{m+1},$$
$$x_{n+1} \geqslant 0.$$

We shall refer to this problem as problem (C). We now give problem $(\tilde{C})$, dual with respect to (C).

Minimize the linear form

$$\bar{L}(Y)=\sum_{i=1}^{m+1} b_i y_i$$

subject to the conditions

$$\sum_{i=1}^{m} a_{ij} y_i + a_{m+1,j} y_{m+1} \geqslant c_j, \qquad j=1, 2, \ldots, n;$$

$$y_{m+1} \geqslant 0.$$

If

$$\sum_{j=1}^{n} a_{m+1,j} x_j^* \leqslant b_{m+1}, \tag{8.20}$$

the vector X^* obviously solves problem (C).

Now let condition (8.20) not hold, i. e.,

$$\sum_{j=1}^{n} a_{m+1,j} x_j^* > b_{m+1}.$$

We shall solve problem (C) by the dual method.

Suppose that problem (A) has been solved by the second simplex or second dual algorithm. In either case, the last principal tableau contains

(a) the optimal program $X^*=(x_1^*, \ldots, x_n^*)$ of the primal problem,

(b) the optimal program $Y^*=(y_1^*, \ldots, y_m^*)$ of the dual problem,

(c) the matrix $\|e_{ij}^*\|_m=(A_{s_1}, A_{s_2}, \ldots, A_{s_m})^{-1}$, the inverse of the optimal-basis matrix.

From the restraints of problem $(\tilde{C})$ we see that the vector

$$\bar{Y}^*=(y_1^*, \ldots, y_m^*, 0)$$

is its support program with the basis $\bar{A}_{s_1}, \ldots, \bar{A}_{s_m}, \bar{A}_{n+1}$. Here, by definition,

$$\bar{A}_j=(a_{1j}, \ldots, a_{mj}, a_{m+1,j}), \qquad j=1, 2, \ldots, n,$$

$$\bar{A}_{n+1}=(0, 0, \ldots, 0, 1).$$

Hence, when solving the problem $(\tilde{C})$ by the dual method, the vector $\bar{Y}^*$ can be taken as the initial support program. The corresponding principal tableau is easily constructed from the last principal tableau of problem (A). It is left to the reader (see Exercise 10) to verify the following relationships:

$$\bar{X}^*=(x_1^*, \ldots, x_n^*, x_{n+1}^*), \tag{8.21}$$

where

$$x_{n+1}^*=-\sum_{j=1}^{n} a_{m+1,j} x_j^* + b_{m+1};$$

$$\bar{Y}^*=(y_1^*, \ldots, y_m^*, 0); \tag{8.22}$$

$$\bar{e}_{ij}^*=e_{ij}^*, \qquad i, j=1, 2, \ldots, m; \tag{8.23}$$

$$\bar{e}_{i,m+1}^*=0, \qquad 1=1, 2, \ldots, m; \tag{8.24}$$

$$\bar{e}_{m+1,j}^*=-\sum_{\lambda=1}^{m} e_{\lambda j}^* a_{m+1,s_\lambda}, \qquad j=1, 2, \ldots, m; \tag{8.25}$$

$$\bar{e}_{m+1,m+1}^*=1. \tag{8.26}$$

The bar indicates that the parameters refer to problem (C).

The optimal program of the complex problem is generally obtained after few iterations of the dual method. This is due to the fact that program $\bar{Y}^*$ is invariably close to the solution of problem $(\tilde{C})$.

8-3. When solving linear-programming problems, we may often essentially reduce the number of iterations if the process of solution can be terminated on a program for which the linear form differs from its optimal value by at most some predetermined value. It is, therefore, necessary to be able to estimate the decrease in the value of the linear form arising when the optimal program is replaced by some intermediate program. This estimate can be obtained by solving the primal and the dual problems simultaneously, or, equivalently, by solving the problem by simplex and dual methods simultaneously.

Consider a pair of dual linear-programming problems. Let $X=(x_1, \ldots, x_n)$ and $Y=(y_1, \ldots, y_m)$ be some support programs of these problems. Let V be the maximum value of the linear form of the primal problem. We know that

$$V \leqslant \tilde{L}(Y) = \sum_{i=1}^{m} b_i y_i.$$

Hence

$$V - L(X) \leqslant \tilde{L}(Y) - L(X). \tag{8.27}$$

Thus, given some program of the dual problem, we may estimate the departure of the linear form of the primal problem on an intermediate program from its optimal value. According to the first duality theorem, the optimal values of the linear forms of the dual and the primal problems coincide. The estimate (8.27) is, therefore, more exact the closer Y is to the solution of the dual problem.

Suppose now that the primal problem is simultaneously solved by two methods, i.e., the simplex method starting with program X, and the dual method starting with program Y. As a result, we obtain two sequences of programs

$$X,\ X_1,\ X_2, \ldots, X_k; \quad Y,\ Y_1,\ Y_2, \ldots, Y_k.$$

From (8.12) we have

$$\delta_k = V - \sum_{j=1}^{n} c_j x_j^{(k)} \leqslant \sum_{i=1}^{m} b_i y_i^{(k)} - \sum_{j=1}^{n} c_j x_j^{(k)} = \tilde{\delta}_k.$$

If δ_k falls within preset limits, the process of solution is terminated and program X_k is taken as the optimum. Otherwise, we should successively pass to new pairs of programs until $\tilde{\delta}_k$ becomes less than the present value.

This *combined method* is particularly effective when applied to linear-programming problems with numerous variables and restraints.

There is still another advantage to the combined method. Assume that the optimal program computed by the simplex method is degenerate. In this case, to establish the optimality of the program we should perform, generally speaking, several iterations. If the problem is solved by the combined method, the optimality of programs of the dual problems is detected directly when the values of the corresponding linear forms coincide.

We now give a geometrical illustration of the combined method.

Example. Maximize the linear form

$$L(X) = 10x_1 + 8x_2 + 7x_3 + 16x_4 + 21x_5$$

subject to the conditions

$$\begin{gathered} 4x_1 + 2x_2 + 5x_3 + 10x_4 + 5x_5 = 6, \\ 9x_1 + 10x_2 + 12.5x_3 + 18x_4 + 16.5x_5 = 14, \\ x_j \geqslant 0, \quad j = 1, 2, \ldots, 5. \end{gathered}$$

S o l u t i o n. In Chapter 4, 4-3, this problem was solved by the simplex method. In 2-8 of the present chapter the same problem was solved by the dual method. In both cases, the solution of the problem was illustrated by geometrical constructions in the three-dimensional space.

Without repeating the considerations of the preceding articles, we shall apply the results in order to plot, on a plane, the successively contracting limits of the deviation of the current values of the linear form from the optimum. It must be kept in mind in the process that the value of the linear form of the dual problem on any program coincides with the value of the linear form of the primal problem on the corresponding pseudo-program.

We cut the polyhedral cone spanned by the augmented restraint vectors of the problem by plane H passing through line Q. All subsequent constructions are made in this plane (Figure 6.9).

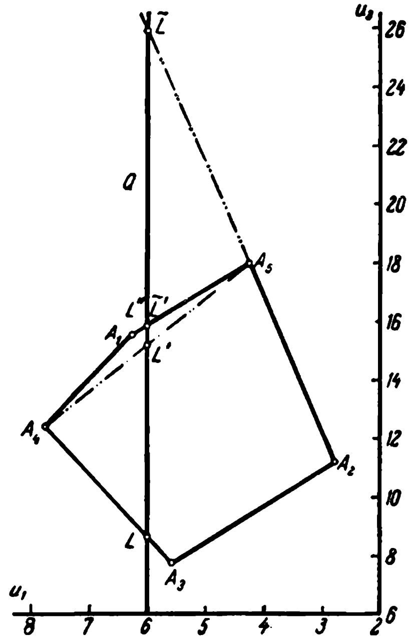


FIGURE 6. 9

The polyhedron $A_4A_1A_5A_2A_3$ is the section of the cone formed by plane H. The vertices A_j of the polyhedron belong to the conical edges corresponding to the augmented restraint vectors $\tilde{A}_j$. The planes Π, Π', Π'' (images of the support programs X, X', X'' of the primal problem) meet plane H along lines A_3A_4, A_5A_4, A_5A_1, respectively. The planes $\tilde{\Pi}$, $\tilde{\Pi}'$ (images of the support programs Y, Y' of the dual problem) meet plane H along lines A_2A_5, A_5A_1, respectively. The ordinates of the intersection points L, L', L'', $\tilde{L}$, $\tilde{L}'$ of these lines with line Q define the values of the linear forms of the primal and the dual problems on the corresponding programs.

We see that the values of the linear form on successive programs of the primal problem increase, whereas the linear form of the dual problem decreases on each successive program. At point $L'' = \tilde{L}'$, corresponding to the optimal programs of the primal and the dual problems, the linear forms of the two problems coincide.

Observe that for $m=2$ this m-dimensional variant of the second geometrical interpretation of a dual pair is simpler and more convenient than the $(m+1)$-dimensional variant which is usually considered.

EXERCISES TO CHAPTER 6

1. Prove necesssity of the optimality test stated in 1-1 for the case when the support program of the dual problem is nondegenerate.

2. Prove that the distance of point $X=(x_1, x_2, \dots, x_n)$, where

$$x_j=\begin{cases} x_{i0}^0-x_t x_{it}^0, & j=s_i, \quad i=1, 2, \dots, m, \\ x_t, & j=t, \\ 0 & \text{in other cases,}\end{cases}$$

from the hyperplane

$$L(X)=L(X_0)$$

(X_0 is a pseudoprogram of problem (1.1)–(1.3) with the basis $A_{s_1}, \dots, A_{s_m}$) is equal to

$$\frac{|x_t \Delta_t|}{\sqrt{\sum_{j=1}^{n} c_j^2}}.$$

3. Prove that problems (3.4)–(3.6) and (3.1)–(3.3) constitute a dual pair.

4. Prove that the vector $Y=(y_1, y_2, \dots, y_m)$ is the kernel of a support program of problem (3.4)–(3.6) if and only if conditions (3.8) are satisfied and among restraint vectors A_j, $j=1, 2, \dots, n$, there are m linearly independent vectors satisfying relationships (3.12).

5. Prove that a support program (Y, Z', Z'') of problem (3.4)–(3.6) is nondegenerate if and only if its kernel Y satisfies the condition

$$\Delta_j=(A_j, Y)-c_j \neq 0, \qquad j \notin I_Y,$$

where I_Y is the set of indices of the basis vectors of the kernel Y of a support program of the dual problem (3.4)–(3.6).

6. Pseudoprogram X of problem (3.1)–(3.3) and kernel Y of a support program of problem (3.4)–(3.6) correspond to each other, if their components are related by (3.14), (3.15). Show that the vector Y corresponding, in this sense, to pseudoprogram X, which is a feasible program of problem (3.1)–(3.3), is the kernel of the optimal program of the dual problem (3.4)–(3.6).

7. Let X be a pseudoprogram of problem (3.1)–(3.3) and Y the corresponding kernel of a program of the dual problem (3.4)–(3.6). Prove that

$$\sum_{j=1}^{n} c_j x_j=\sum_{i=1}^{m} b_i y_i-\sum_{j=1}^{n} \Delta_j \gamma_j,$$

where

$$\gamma_j\begin{cases} \alpha_j & \text{for} \quad \Delta_j \geqslant 0, \\ \beta_j & \text{for} \quad \Delta_j < 0.\end{cases}$$

8. Let Y be a degenerate support program of the dual problem (1.4)–(1.5) with $m+\nu$ parameters Δ_j equal to zero. Construct an example of a problem with $C_{m+\nu}^{\nu}$ bases of support program Y.

9. Let $A_{s_1}, A_{s_2}, \dots, A_{s_m}$ be a system of linearly independent restraint vectors of problem (1.1)–(1.3). We limit the variables x_j, whose parameters $\Delta_j=\sum_{i=1}^{m} c_{s_i} x_{ij}-c_j$ are negative, by the number $M>0$ and solve the resulting bounded problem by the dual method. Let E be the set of indices of the extrabasis variables of the solution of the bounded problem. Prove the following two propositions:

(a) there exists a number $M_1<\infty$ such that for all $M>M_1$, with $\sum_{j \in E} \Delta_j \neq 0$, the linear form of problem (1.1)–(1.3) is unbounded in its domain of definition;

(b) there exists a number $M_2<\infty$ such that for all $M>M_2$ unsolvability of the bounded problem implies unsolvability of the initial problem (1.1)–(1.3).

10. Verify the validity of relationships (8.21)–(8.26) introduced in 8-2 in connection with the possibility of solving several linear-programming problems differing in a single restraint.

11. Applying the technique given in 7-6, 7-7, determine the initial support program of the problem which is dual with respect to the problem requiring maximization of the linear form

$$L(X)=4x_1+3x_2+10x_4-2x_5$$

subject to the conditions

$$\begin{aligned} &3x_1+2x_2-x_3+5x_4-2x_5=8, \\ &\quad x_2+3x_3+6x_4+3x_5=15, \\ &2x_1 \quad -x_3+x_4-2x_5=0, \\ &x_j \geqslant 0, \quad j=1, 2, 3, 4, 5. \end{aligned}$$

12. Solve the problem in Exercise 11 following the first and the second dual algorithms.

13. Given the optimal program of the problem in Exercise 11, compute the components of the solution of the dual problem.

14. Solve the problem in Exercise 11 subject to the additional restraints

$$x_j \leqslant 2, \quad j=1,\ 2,\ 3,\ 4,\ 5.$$

15. Draw a block diagram of the gradient method for computing a support program of a linear-programming problem from any feasible program.

16. A linear-programming problem calls for the maximization of the linear form

$$L(X) = -(a+1)x_1 + (a-b)x_2 + (a+b-c-1)x_3 + (b+c-2)x_4 + cx_5 + x_6 + x_7 + x_8$$

subject to the conditions

$$\begin{aligned} x_1 + x_2 + ax_3 + bx_4 + cx_5 + x_6 \qquad &= 10, \\ -x_1 + ax_2 + bx_3 + cx_4 + c_5 \qquad + x_7 &= 5, \\ ax_1 + bx_2 + cx_3 + x_4 + x_5 \qquad + x_8 &= 20, \\ x_j \geqslant 0. \quad j = 1,\ 2,\ 3,\ 4,\ 5. \end{aligned}$$

Determine the range of the parameters a, b, and c in which pseudoprogram $X=(0,\ 0,\ 0,\ 0,\ 0,\ 10,\ 5, -20)$ leads to cases (a), (b), and (c), respectively.

Chapter 7

THE HUNGARIAN METHOD

The simplex method is applied to the primal problem only and the dual simplex method deals only with the dual problem. We will now discuss a method which deals with both problems of the dual pair. In the method of successive residue contraction [known as the Hungarian method] the optimal program is found by moving over vectors with nonnegative components. The rules of transition from one vector X to another are provided by contracting, from step to step, the differences between the right- and the left-hand sides of the quality restraints of the primal problem written in the canonical form. These differences are called residues, and the Hungarian method is thus referred to as the method of successive residues. It is proved that after a finite number of steps these difference will vanish or unsolvability of the problem will be established.

The principal concepts of this method were first advanced in 1939 by L. V. Kantorovich, who used the method to solve particular linear-programming problems /61, 67/. L. V. Kantorovich does not consider the dual problem, but using decision multipliers (see Chapter 3, 5-1) defines its optimal program.

An independent description of this method was given in 1956 by three American authors /52/. The history of this method in the American literature is as follows. First, on the basis of some preliminary considerations contained in one of the older works of the Hungarian mathematician Egerváry /123/, the dual simplex method was extended for the transportation problem /70, 111, 76/. In honor of Egerváry the method was then called the Hungarian method. In /52/ the Hungarian method for the solution of the transportation problem was extended to the case of the general linear-programming problem.

The material in this chapter is presented according to the following outline.

In § 1 we describe the basic concepts of the method and give general considerations concerning the computational procedure. In § 2 the theoretical principles of the method are illustrated by two simple examples. In § 3 we give a geometrical interpretation of the Hungarian method. In § 4 the method is extended to linear-programming problems with bilateral restraints. The algorithm of the method is disccused in § 5. Here the computational procedure of the Hungarian method is presented in application to problems in canonical form and problems with bilateral restraints and the discussion is elucidated by suitable examples.

In the last section, § 6, we discuss the revised Hungarian method— the

method of bilateral evaluations. This revised method often makes it possible to obtain and evaluate an approximate solution of a linear-programming problem after appreciably fewer computations than in all other methods.

§ 1. General outline

1-1. We write the linear-programming problem in canonical form. Maximize the linear form

$$L(X)=\sum_{j=1}^{n} c_j x_j \tag{1.1}$$

subject to the conditions

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \qquad i=1, 2, \ldots, m, \tag{1.2}$$

$$x_j \geqslant 0, \qquad j=1, 2, \ldots, n. \tag{1.3}$$

The primal problem (1.1)–(1.3) will be denoted by (A). Problem $(\tilde{A})$ dual to problem (A) is to minimize the linear form

$$\tilde{L}(Y)=\sum_{i=1}^{m} b_i y_i \tag{1.4}$$

subject to the conditions

$$\sum_{i=1}^{m} a_{ij} y_i \geqslant c_j, \qquad j=1, 2, \ldots, n. \tag{1.5}$$

We shall assume all the components of the constraint vector $B=(b_1, b_2, \ldots, b_m)^T$ to be nonnegative. Clearly, every linear-programming problem can be reduced to an equivalent problem satisfying this condition.

The discussion of the method is simplified if we introduce the so-called *augmented* and *auxiliary problems*.

For brevity, we shall denote the augmented problem by (B). Problem (B) is to minimize the linear form

$$\sum_{i=1}^{m} \varepsilon_i \tag{1.6}$$

subject to the conditions

$$\sum_{j=1}^{n} a_{ij} x_j + \varepsilon_i = b_i, \qquad i=1, 2, \ldots, m, \tag{1.7}$$

$$x_j \geqslant 0, \qquad j=1, 2, \ldots, n, \tag{1.8}$$

$$\varepsilon_i \geqslant 0, \qquad i=1, 2, \ldots, m. \tag{1.9}$$

Let $Y=(y_1, y_2, \ldots, y_m)$ be a feasible program of the dual problem $(\tilde{A})$ (not necessarily a support program). The methods for finding an initial program of the dual problem are discussed in Chapter 6, 7-2.

We select the restraint vectors A_j, $j=1, 2, \ldots, n$, such that

$$\sum_{i=1}^{m} a_{ij} y_i = c_j. \tag{1.10}$$

The set of indices of these vectors is denoted by E_Y. To each program Y of the dual problem $(\tilde{A})$ we can associate an auxiliary problem (C_Y) which is obtained from the augmented problem (B) when additional restraints are imposed:

$$x_j = 0 \quad \text{for} \quad j \notin E_Y. \tag{1.11}$$

The auxiliary problem (C_Y) is, thus, to minimize the linear form

$$\sum_{i=1}^{m} \varepsilon_i \tag{1.12}$$

subject to the conditions

$$\sum_{j \in E_Y} a_{ij} x_j + \varepsilon_i = b_i, \tag{1.13}$$

$$x_j \geqslant 0 \quad \text{for} \quad j \in E_Y, \tag{1.14}$$

$$\varepsilon_i \geqslant 0, \qquad i = 1, 2, \ldots, m. \tag{1.15}$$

When solving problem (A) by the Hungarian method we shall require the dual of the auxiliary problem (C_Y) also. We shall denote the dual auxiliary problem by $(\tilde{C}_Y)$. According to the general rules of construction of dual problems, problem $(\tilde{C}_Y)$ is stated as follows.

Maximize the linear form

$$\sum_{i=1}^{m} b_i \mu_i, \tag{1.16}$$

subject to the conditions

$$\sum_{i=1}^{m} a_{ij} \mu_i \leqslant 0 \quad \text{for} \quad j \in E_Y, \tag{1.17}$$

$$\mu_i \leqslant 1, \qquad i = 1, 2, \ldots, m. \tag{1.18}$$

The restraint vectors of the auxiliary problem (C_Y) are the vectors $A_{j_1}, A_{j_2}, \ldots, A_{j_s}, e_1, e_2, \ldots, e_m$ where the indices $j_1, \ldots, j_s$ constitute the set E_Y, and the vectors $e_1, e_2, \ldots, e_m$ form the complete set of m-dimensional unit vectors.

Let the components of a feasible program of the auxiliary problem (C_Y) be $\xi_j (j \in E_Y)$, $\varepsilon_i (i = 1, 2, \ldots, m)$. To every program let there correspond some n-dimensional vector $X = (x_1, \ldots, x_n)$, where

$$\left. \begin{array}{ll} x_j = \xi_j & \text{for} \quad j \in E_Y, \\ x_j = 0 & \text{for} \quad j \notin E_Y. \end{array} \right\} \tag{1.19}$$

Generally speaking, vector X with nonnegative components defined by (1.19) is not a program of primal problem (A). From (1.13), (1.15), (1.19),

$$b_i - \sum_{j=1}^{n} a_{ij} x_j = \varepsilon_i \geqslant 0. \tag{1.20}$$

The components ε_i of a feasible program of the auxiliary problem thus define the residues of the i-th restraint in (1.2) when the vector $X = (x_1, \ldots, x_n)$ is substituted in (1.20). The optimal support program of the auxiliary problem has at most m positive components (ξ_j, ε_i). The number of positive components of the corresponding vector X is a fortiori at most m.

We shall call the vector X corresponding to the optimal support program of some auxiliary problem (C_Y) a q u a s i p r o g r a m of primal problem (A):

The basis vectors of the solution of the auxiliary program form the quasiprogram basis of problem (A). The basis B_X of the quasiprogram is constructed of some restraint vectors of problem (A) and some unit vectors. If the quasiprogram basis contains no unit vectors, the quasiprogram is a feasible program of problem (A) and, as we shall see in what follows, also its solution.

The vector $E=(\varepsilon_1, \ldots, \varepsilon_m)$, where $\varepsilon_1, \ldots, \varepsilon_m$ are the last m components of the solution of the auxiliary problem, will be called the residue vector corresponding to quasiprogram X, and the sum $\varepsilon_0=\sum_{i=1}^{m}\varepsilon_i$ of the components of vector E will be called the residue of the quasiprogram.

Optimality test. A quasiprogram X of problem (A) is its optimal program, if all the components of the corresponding residue vector are zero, or, equivalently, if the quasiprogram with zero residue solves the problem.

Proof. By construction, the components x_j of the quasiprogram are nonnegative. Moreover, by definition

$$\varepsilon_i=0 \quad \text{for} \quad i=1, 2, \ldots, m.$$

Therefore, according to (1.20), the components of the quasiprogram satisfy restraints (1.2) of problem (A). Hence, the quasiprogram is a feasible program of problem (A). According to the definition of program $X=(x_1, \ldots, x_n)$, for $x_j>0$ we have

$$\sum_{i=1}^{m} a_{ij}y_i=c_j.$$

We, therefore, have the following:

$$\sum_{j=1}^{n} c_jx_j=\sum_{j=1}^{n} x_j\sum_{i=1}^{m} a_{ij}y_i=\sum_{i=1}^{m} y_i\sum_{j=1}^{n} a_{ij}x_j=\sum_{i=b}^{m} b_iy_i.$$

By virtue of Chapter 3, Lemma 1.2, this result proves the optimality of program X of problem (A). Hence, program Y of dual problem $(\tilde{A})$ generating the auxiliary problem (C_Y) is the solution of problem $(\tilde{A})$. This completes the proof.

1-2. Let $Y=(y_1, \ldots, y_m)$ be a feasible program of the dual problem $(\tilde{A})$. Let auxiliary problem (C_Y) and dual auxiliary problem $(\tilde{C}_Y)$ be associated with it.

Let the optimal program of problem (C_Y) have the form

$$(\Xi^*, E^*)=(\xi^*_{j_1}, \ldots, \xi^*_{j_s}, \varepsilon^*_1, \ldots, \varepsilon^*_m),$$

and the solution of problem $(\tilde{C}_Y)$ be the vector

$$M^*=(\mu^*_1, \ldots, \mu^*_m).$$

We introduce the parameters Δ_j and δ^*_j, $j=1, 2, \ldots, n$:

$$\Delta_j=\sum_{i=1}^{m} a_{ij}y_i-c_j, \qquad j=1, 2, \ldots, n, \tag{1.21}$$

$$\delta^*_j=\sum_{i=1}^{m} a_{ij}\mu^*_i, \qquad j=1, 2, \ldots, n. \tag{1.22}$$

The solution of auxiliary problem (C_Y) determines a quasiprogram X of problem (A). The residue vector E^* and the parameter δ^*_j are the defining characteristics of quasiprogram X. Depending on the residues ε^*_i and the

signs of the parameters δ_j^*, we have three cases:

(a) All the components of the residue vector are zero:

$$\varepsilon_i^* = 0, \quad i = 1, 2, \ldots, m.$$

(b) There are positive residues and all the parameters δ_j^* are nonpositive:

$$\varepsilon_0^* = \sum_{i=1}^{m} \varepsilon_i^* > 0, \quad \delta_j^* \leqslant 0, \qquad j = 1, 2, \ldots, n.$$

(c) The residue vectors have some positive components and at least one of the δ_j^* is positive.

In case (a), it follows from the optimality test that the quasiprogram is also the optimal program of problem (A). We shall show in the following that in case (b) problem (A) in unsolvable (problem restraints are inconsistent), and in case (c) we may pass to a new quasiprogram with a smaller residue.

1-3. To simplify the analysis of cases (b) and (c), we will note the variation of the sum of residues (the linear form of the auxiliary problem) when passing from program Y of the dual problem to the program

$$Y' = Y(\theta) = Y - \theta M^*.$$

The transition from program Y to program $Y(\theta)$ is an e l e m e n t a r y t r a n s f o r m a t i o n of program Y.

We compute the parameters $\Delta_j(\theta)$ for program $Y(\theta)$. From (1.21) and (1.22) we have

$$\Delta_j(\theta) = \sum_{i=1}^{m} a_{ij} y_i(\theta) - c_j = \Delta_j - \theta\delta_j^*, \qquad j = 1, 2, \ldots, n. \tag{1.23}$$

The vector $Y(\theta)$ is a feasible program of the dual problem $(\tilde{A})$ for all θ such that

$$\Delta_j(\theta) \geqslant 0 \text{ for } j = 1, 2, \ldots, n.$$

We compute the value of the linear form of problem $(\tilde{A})$ on feasible program $Y(\theta)$. We have

$$\tilde{L}[Y(\theta)] = \sum_{i=1}^{m} b_i y_i(\theta) = \sum_{i=1}^{m} b_i y_i - \theta \sum_{i=1}^{m} b_i \mu_i^*.$$

Applying the first duality theorem to the auxiliary problem (C_Y) and to its dual problem $(\tilde{C}_Y)$, we obtain

$$\sum_{i=1}^{m} b_i \mu_i^* = \sum_{i=1}^{m} \varepsilon_i^*.$$

Hence,

$$\tilde{L}[Y(\theta)] = \tilde{L}(Y) - \theta \sum_{i=1}^{m} \varepsilon_i^*. \tag{1.24}$$

We see that if $\theta > 0$ and $\varepsilon_0^* = \sum_{i=1}^{m} \varepsilon_i^* > 0$, the linear form of the dual problem decreases under elementary transformation of program Y into program $Y(\theta)$.

We shall now analyze case (b), where some of the residues ε_i^* are positive and all $\delta_j^* \leqslant 0$, $j = 1, 2, \ldots, n$. In this case we see from (1.23) that $Y(\theta)$ is a feasible program of problem $(\tilde{A})$ for any $\theta \geqslant 0$. Equality (1.24) shows that problem $(\tilde{A})$ is not bounded below in the set of its feasible programs. Hence, according to Chapter 3, Lemma 1.3, the primal problem (A) is unsolvable

since its restraints are inconsistent. The conditions determining case (b) will be referred to in what follows as the unsolvability test of the problem.

We now proceed to analyze case (c), where some of the residues ε_i^* are positive and also some of the parameters δ_j^* are positive.

The greatest θ for which $Y(\theta)$ is still a feasible program of problem $(\tilde{A})$ is found from

$$\theta_0 = \min_{\delta_j^* > 0} \left(\frac{\Delta_j}{\delta_j^*} \right). \tag{1.25}$$

Here the minimum is taken over the indices j for which δ_j^* are positive. In case (c) there is at least one such index. Applying the definition of δ_j^* and conditions (1.17) satisfied by μ_i^*, we obtain

$$\delta_j^* \leqslant 0, \quad j \in E_Y.$$

Hence θ_0 is obtained for $j \notin E_Y$. On the other hand, from the definition of the set E_Y the parameters Δ_j are positive for $j \notin E_Y$. Therefore, $\theta_0 > 0$.

We shall show that program $Y' = Y(\theta_0)$ of problem $(\tilde{A})$ leads to auxiliary problem $(C_{Y'})$ with a smaller minimum sum of residues than for problem (C_Y).

From the second duality theorem, for

$$\xi_{j_\lambda}^* > 0$$

we have

$$\delta_{j_\lambda}^* = \sum_{i=1}^{m} a_{ij_\lambda} \mu_i^* = 0.$$

From the preceding relationships we see that $E_{Y'}$ comprises, in particular, the subscripts j such that $x_j = \xi_j^* > 0$. Indeed, for these j

$$\Delta_j(\theta) = \Delta_j - \theta_0 \delta_j^* = 0 - \theta_0 \cdot 0 = 0.$$

The initial program of auxiliary problem $(C_{Y'})$ corresponding to $Y(\theta_0)$ can, therefore, be constructed from the positive components of the optimal program of the previous auxiliary problem (C_Y). This essentially reduces the number of steps leading to the solution of the current auxiliary problem.

We define j_0 from the relationship

$$\theta_0 = \frac{\Delta_{j_0}}{\delta_{j_0}^*}, \quad \delta_{j_0}^* > 0.$$

There may be several such indices. Clearly, vector A_{j_0} is one of the restraint vectors of the new auxiliary problem, since

$$\Delta_{j_0}(\theta_0) = \Delta_{j_0} - \theta_0 \delta_{j_0}^* = 0.$$

Let problem C_Y' be nondegenerate (to this end it is sufficient to assume nondegeneracy of the augmented problem (B)).

In § 5 we discuss when it is expedient to solve the auxiliary problem by the second simplex algorithm. According to the basic premises of this method, the condition $\delta_{j_0}^* > 0$ for a nondegenerate problem indicates that the introduction of vector A_{j_0} into the basis of the initial program of the auxiliary program will decrease its linear form. The minimum of the new auxiliary problem is, therefore, strictly less than the minimum of the

previous auxiliary problem:

$$\sum_{i=1}^{m} \varepsilon_i^{*\prime} = \sum_{i=1}^{m} \varepsilon_i^*(\theta_0) < \sum_{i=1}^{m} \varepsilon_i^*.$$

Thus, in case (c) we can find an elementary transformation of program Y of the dual problem into program $Y' = Y(\theta_0)$ which generates an auxiliary problem $(C_{Y'})$ with a lower minimum sum of residues. In other words, in case (c) we may pass from quasiprogram X to quasiprogram X' with a lower residue. This sequence of operations constitutes one iteration of the Hungarian method. The procedure is repeated until we obtain either case (a) or case (b).

1-4. We outline briefly the sequence of operations in one iteration of the Hungarian method.

In every iteration we know the program Y of the dual problem $(\tilde{A})$ (this having been obtained at the end of the previous iteration). The iteration starts with the construction of auxiliary problem (C_Y). The solution of problem (C_Y) gives a quasiprogram X of the primal problem (A). Quasiprogram X is then investigated. From the magnitude of the quasiprogram residue and the signs of the parameters δ_i^* we establish which of the three cases ((a), (b), or (c)) applies.

If the conditions of the optimality test are satisfied, i.e., case (a), the quasiprogram obtained is the optimal program of problem (A) and the process of solution is terminated. If the conditions of the optimality test are not fulfilled, we must examine the signs of the parameters δ_j^*. If all δ_j^* are nonpositive, the problem is unsolvable (case (b)). If some of the δ_j^* are positive, case (c) applies and a new program Y' of the dual problem, which is used in a successive iteration reducing the quasiprogram residue, is constructed.

Each iteration of the Hungarian method is made up of several simplex iterations required to solve the corresponding auxiliary problem. We observe that the feasible programs of any auxiliary problem are also programs of the augmented problem (B). The process of solution of the auxiliary problems thus involves successive transitions from one support program of the augmented problem to another.

Transitions of support programs of problem (B) follow the simplex rules. This, however, does not mean that the solution of problem (A) by the Hungarian method involves determining the optimal program of problem (B) by the simplex method. Indeed, throughout every iteration of the Hungarian method the vector to be introduced in the basis is chosen not from the entire set of restraint vectors of problem (B), but only from among the unit vectors

$$e_j \ (j = 1, 2, \ldots, m)$$

and those A_j for which

$$\Delta_j = \sum_{i=1}^{m} a_{ij} y_i - c_j = 0.$$

Here $Y = (y_1, \ldots, y_m)$ is a feasible program of problem $(\tilde{A})$ associated with the current iteration. Before starting with each iteration, the set of admissible vectors A_j is restored using the new program of problem $(\tilde{A})$.

The preceding considerations make it possible, in particular, to draw some conclusions concerning the finiteness of the Hungarian method. Indeed, for a nondegenerate augmented problem the transition from one support

program to another is accompanied by monotonic decrease of linear form (1.6). Therefore, in the process of solution we cannot return to a support program of problem (B) which has already been examined. The number of different support programs of the augmented problem is finite. Hence, the optimal program of a solvable problem (A) is obtained by the Hungarian method after a finite number of iterations. If the problem is unsolvable, its unsolvability is also established in a finite number of steps.

If degenerate auxiliary problems are solved using the rules ensuring against cycling (see Chapter 5, 5-5), a degenerate augmented problem is also solved by the Hungarian method in a finite number of steps.

§ 2. Examples

2-1. We shall now illustrate the above by the example investigated in Chapter 4, § 3, by the simplex method and in Chapter 6, § 2, by the dual simplex method.

Example 1. Maximize the linear form

$$L(X)=5x_1-x_2-2x_3+5x_4+5x_5-x_6$$

subject to the conditions

$$\begin{aligned}
-2x_1+5x_2+x_3 &=10,\\
x_1-x_2+x_4 &=1,\\
x_1+2x_2+x_5 &=6,\\
10x_1-3x_2+x_6 &=15,\\
x_j\geqslant 0,\quad j=1,\,2,\,\ldots,\,6.&
\end{aligned}$$

Solution. The above problem will be called problem (A). Its dual, problem $(\tilde{A})$, is stated as follows: Minimize the linear form

$$\tilde{L}(Y)=10y_1+y_2+6y_3+15y_4$$

subject to the condtions

$$\begin{aligned}
-2y_1+y_2+y_3+10y_4 &\geqslant 5,\\
5y_1-y_2+2y_3-3y_4 &\geqslant -1,\\
y_1 &\geqslant -2,\\
y_2 &\geqslant 5,\\
y_3 &\geqslant 5,\\
y_4 &\geqslant -1.
\end{aligned}$$

Problem (A) corresponds to the following augmented problem (B): Minimize the linear form

$$\varepsilon_1+\varepsilon_2+\varepsilon_3+\varepsilon_4$$

subject to the conditions

$$\begin{aligned}
-2x_1+5x_2+x_3+\varepsilon_1 &=10,\\
x_1-x_2+x_4+\varepsilon_2 &=1,\\
x_1+2x_2+x_5+\varepsilon_3 &=6,\\
10x_1-3x_2+x_6+\varepsilon_4 &=15,\\
x_j\geqslant 0,\quad j=1,\,2,\,\ldots,\,6,&\\
\varepsilon_i\geqslant 0,\quad i=1,\,2,\,3,\,4.&
\end{aligned}$$

It is easy to see that

$$Y=(0,\,5,\,5,\,0)$$

is a feasible program of problem $(\tilde{A})$. We compute the parameters Δ_j, i.e., the differences between the left- and the right-hand sides of restraints of problem $(\tilde{A})$. We have

$$\Delta=(\Delta_1,\,\Delta_2,\,\Delta_3,\,\Delta_4,\,\Delta_5,\,\Delta_6)=(5,\,6,\,2,\,0,\,0,\,1).$$

The set E_Y in this case contains two indices: $j=4$ and $j=5$ $(E_Y=\{4,\,5\})$.

Auxiliary problem (C_Y) generated by program Y is obtained from problem (B) when we set $x_j=0$ for $j\notin E_Y$. In problem (C_Y) we thus minimize

$$\sum_{i=1}^{4}\varepsilon_i$$

subject to the conditions

$$\begin{array}{rl} \varepsilon_1 & =10, \\ x_4 \quad +\varepsilon_2 & =1, \\ x_5 \quad\quad +\varepsilon_3 & =6, \\ \varepsilon_4 & =15, \end{array}$$
$$x_4 \geqslant 0, \quad x_5 \geqslant 0, \quad \varepsilon_i \geqslant 0, \quad i=1, 2, 3, 4.$$

Without special computations, we see that the solution of problem (C_Y) is given by the following set of numbers:

$$x_4=1, \quad x_5=6, \quad \varepsilon_1=10, \quad \varepsilon_2=0, \quad \varepsilon_3=0, \quad \varepsilon_4=15.$$

Thus,

$$(\Xi^*, E^*)=(1, 6, 10, 0, 0, 15).$$

The initial quasiprogram is thus the vector $X=(0,0,0,1,6,0)$. The residue of quasiprogram X does not vanish $\left(\sum_{i=1}^{4} \varepsilon_i^* = 10+0+0+15=25\right)$. Case (a) does not apply: quasiprogram X does not solve the problem.

Problem $(\tilde{C}_Y)$, the dual of problem (C_Y), is to maximize the linear form

$$10\mu_1+\mu_2+6\mu_3+15\mu_4$$

subject to the conditions

$$\mu_2 \leqslant 0, \quad \mu_3 \leqslant 0, \quad \mu_i \leqslant 1, \quad i=1, 2, 3, 4.$$

Since all the coefficients of the linear form of problem $(\tilde{C}_Y)$ are positive, problem $(\tilde{C}_Y)$ is solved by the vector

$$M^*=(1, 0, 0, 1).$$

Following the recommended procedure, we find the parameters

$$\delta_j^*=\sum_{i=1}^{4} a_{ij}\mu_i^*$$

for all $j\ (j=1, 2, \ldots, 6)$. We have

$$\delta^*=(\delta_1^*, \delta_2^*, \delta_3^*, \delta_4^*, \delta_5^*, \delta_6^*)=(8, 2, 1, 0, 0, 1).$$

Some of the components δ_j^* of the vector δ^* are positive. We thus have case (c).

We compute

$$\theta_0=\min_{\delta_j^*>0} \frac{\Delta_j}{\delta_j^*}.$$

We have

$$\theta_0=\min\left(\frac{5}{8}, 3, 2, 1\right)=\frac{5}{8}.$$

The vector $\Delta'=\Delta(\theta_0)$ is computed from

$$\Delta'=\Delta-\theta_0\delta^*.$$

In our case,

$$\Delta'=\left(0, \frac{19}{4}, \frac{11}{8}, 0, 0, \frac{3}{8}\right).$$

We form a successive program Y' of problem $(\tilde{A})$:

$$Y'=Y(\theta_0)=Y-\theta_0 M^*,$$
$$Y'=\left(-\frac{5}{8}, 5, 5, -\frac{5}{8}\right).$$

Program Y' of problem $(\tilde{A})$ generates a new auxiliary problem $(C_{Y'})$ which is to minimize the linear form

$$\sum_{i=1}^{4} \varepsilon_i$$

subject to the conditions

$$\begin{array}{rl} -2x_1 \quad\quad +\varepsilon_1 & =10, \\ x_1+x_4 \quad\quad +\varepsilon_2 & =1, \\ x_1 \quad +x_5 \quad\quad +\varepsilon_3 & =6, \\ 10x_1 \quad\quad +\varepsilon_4 & =15, \end{array}$$
$$x_1 \geqslant 0, x_4 \geqslant 0, \quad x_5 \geqslant 0; \quad \varepsilon_i \geqslant 0, \quad i=1, 2, 3, 4.$$

We solve problem $(C_{Y'})$ by the second simplex algorithm. It is natural to use the solution of problem (C_Y) as the initial program of problem $(C_{Y'})$. Already after the first iteration we find the optimal programs of problems $(C_{Y'})$ and $(\tilde{C}_{Y'})$:

$$(\Xi^{*\prime}, E^{*\prime})=(1, 0, 5, 12, 0, 0, 5),$$
$$M^*=(1, -8, 0, 1).$$

The corresponding quasiprogram is

$$X'=(1, 0, 0, 0, 5, 0).$$

Its residue is

$$\varepsilon_0^{*\prime}=\sum_{i=1}^{4} \varepsilon_i^{*\prime}=12+0+0+5=17.$$

To establish whether the problem can be considered unsolvable, we determine the signs of the components of the vector $\delta^{*\prime}$. In our case

$$\delta^{*\prime}=(0, 10, 1, -8, 0, 1).$$

We again have case (c), when an elementary transformation can be found to obtain a new program Y'' of problem $(\tilde{A})$ and, consequently, a new quasiprogram of problem (A) with a lower residue. The transformation parameter θ_0' is

$$\theta_0'=\min\left(\frac{19}{40}, \frac{11}{8}, \frac{3}{8}\right)=\frac{3}{8}.$$

We compute the vectors

$$\Delta''=\Delta'(\theta_0'), \quad Y''=Y'(\theta_0')$$

and form the successive auxiliary problem $(C_{Y''})$.

Subsequent computations follow the same procedure. The main results of each iteration are given in Table 7.1.

After the fifth iteration, we obtain the quasiprogram

$$X^{(V)}=(48/23, 45/23, 101/23, 20/23, 0, 0)$$

with a zero residue. According to the optimality test, $X^{(V)}$ solves problem (A). The maximum value of linear form $L(X)$ is

$$L(X^{(V)})=93/23.$$

It is readily seen that $Y^{(V)}$ is the optimal program of problem $(\tilde{A})$. Indeed,

$$L(Y^{(V)})=L(X^{(V)})=93/23.$$

2-2. We now consider another example, which was described in Chapter 6, 2-6, to illustrate case (b) of an unsolvable linear-programming problem.

Example 2. Maximize the linear form

$$L(X)=5x_1+4x_2+x_5$$

subject to the conditions

$$\begin{aligned}
x_1+2x_2-x_3 \qquad\qquad &= 2,\\
4x_1+3x_2 \qquad +x_4 \qquad &=12,\\
4x_1+4x_2+x_3+x_4+x_5 &=10,\\
x_j\geqslant 0 \quad j=1, 2, 3, 4, 5.&
\end{aligned}$$

Solution. The above problem will be called problem (A). The dual problem $(\tilde{A})$ is to minimize the linear form

$$\tilde{L}(Y)=2y_1+12y_2+10y_3$$

subject to the conditions

$$\begin{aligned}
y_1+4y_2+4y_3&\geqslant 5,\\
2y_1+3y_2+4y_3&\geqslant 4,\\
-y_1 \qquad + y_3&\geqslant 0,\\
y_2+y_3&\geqslant 0,\\
y_3&\geqslant 1.
\end{aligned}$$

As an initial program of problem $\tilde{A}$ we naturally take the vector $Y=(0, 0, y_3)$, where

$$y_3=\max\{5/4, 1, 0, 0, 1\}=5/4.$$

TABLE 7. 1

No. of itera-tion (ν)	i	$X^{(\nu)}$	$\Delta^{(\nu)}$	$\delta^{*(\nu)}$	l	1	2	3	4
0	1		5	8	Y		5	5	
	2		6	2	E^*	10			15
	3		2	1	M^*	1			1
	4	1			B_X	A_4	A_5	e_1	e_4
	5	6			$\Sigma\, \varepsilon_l^*$	25			
	6		1	1	θ_0	$5/8$			
1	1	1			Y'	$-5/8$	5	5	$-5/8$
	2		$19/4$	10	$E^{*\prime}$	12			5
	3		$11/8$	1	$M^{*\prime}$	1	-8		1
	4			-8	$B_{X'}$	A_1	A_5	e_1	e_4
	5	5			$\Sigma\, \varepsilon_l^{*\prime}$	17			
	6		$3/8$	1	θ_0'	$3/8$			
2	1	1			Y''	-1	8	5	-1
	2		1	3	$E^{*\prime\prime}$	12			
	3		1	1	$M^{*\prime\prime}$	1	2		
	4		3	2	$B_{X''}$	A_1	A_5	e_1	A_6
	5	5			$\Sigma\, \varepsilon_l^{*\prime\prime}$	12			
	6	5			θ_0''	$1/8$			

TABLE 7. 1 (continued)

No. of iteration(ν)	j \	$X^{(\nu)}$	$\Delta^{(\nu)}$	$\delta^{*(\nu)}$	\ i	1	2	3	4
3	1	$12/7$			Y'''	$-4/3$	$22/3$	5	-1
	2	$5/7$			$E^{*'''}$	$69/7$			
	3		$2/3$	1	$M^{*'''}$	1	$44/7$		$-3/7$
	4		$7/3$	$44/7$	$\mathrm{B}_{X'''}$	A_1	A_5	e_1	A_2
	5	$20/7$			$\Sigma e_i^{*'''}$	$69/7$			
	6			$-3/7$	θ_0'''	$49/132$			
4	1	$43/23$			$Y^{(IV)}$	$-75/44$	5	5	$-37/44$
	2	$45/23$			$E^{*(IV)}$	$101/23$			
	3		$13/44$	1	$M^{*(IV)}$	1		$-44/23$	$9/23$
	4	$20/23$			$\mathrm{B}_{X(IV)}$	A_1	A_4	e_1	A_2
	5			$-44/23$	$\Sigma e_i^{*(IV)}$	$101/23$			
	6		$7/44$	$9/23$	$\theta_0^{(IV)}$	$13/44$			
5	1	$48/23$			$Y^{(V)}$	-2	5	$128/23$	$-22/23$
	2	$45/23$			$E^{*(V)}$				
	3	$101/23$			$M^{*(V)}$				
	4	$20/23$			$\mathrm{B}_{X(V)}$	A_1	A_4	A_3	A_2
	5		$13/23$		$\Sigma \varepsilon_i^{*(V)}$	0			
	6		$1/23$						

2154

For program Y we have

$$\Delta=(0,\ 1,\ 5/4,\ 5/4,\ 1/4).$$

Hence, auxiliary problem (C_Y) contains only one restraint vector of problem (A), namely vector A_1. Problem (C_Y) is to minimize

$$\sum_{i=1}^{3} \varepsilon_i$$

subject to the conditions

$$\begin{gathered} x_1+\varepsilon_1=\ 2, \\ 4x_1+\varepsilon_2=12, \\ 4x_1+\varepsilon_3=10, \\ x_1\geqslant 0,\quad \varepsilon_i\geqslant 0,\qquad i=1,\ 2,\ 3. \end{gathered}$$

It is easily seen, without special computations, that problem (C_Y) is solved by the vector

$$(\Xi^*,\ E^*)=(2,\ 0,\ 4,\ 2).$$

Hence the initial quasiprogram X is a vector with one nonzero component,

$$X=(2,\ 0,\ 0,\ 0,\ 0).$$

The residue of quasiprogram X is 6.

Problem $(\tilde{C}_Y)$ dual to auxiliary problem (C_Y) is solved by the vector

$$M^*=(-8,\ 1,\ 1).$$

We now compute the parameters of quasiprogram X required for passing to the next quasiprogram with a lower residue. We have

$$\begin{gathered} \delta^*=(0,\ -9,\ 9,\ 2,\ 1). \\ \theta_0=\min\frac{\Delta_j}{\delta_j^*}=\frac{5}{36}, \\ \Delta'=\Delta(\theta_0)=(0,\ 9/4,\ 0,\ 35/36,\ 1/9), \\ Y'=(10/9,\ -5/36,\ 10/9). \end{gathered}$$

The program Y' of problem $(\tilde{A})$ generates auxiliary problem $(C_{Y'})$. The solution $(\Xi^*,\ E^*)$ of problem (C_Y) can be taken as the initial program of problem $(C_{Y'})$. After the first iteration we find

$$\begin{gathered} (\Xi^{*\prime},\ E^{*\prime})=(12/5,\ 2/5,\ 0,\ 12/5,\ 0), \\ M^{*\prime}=(-4/5,\ 1,\ -4/5), \\ X'=(12/5,\ 0,\ 2/5,\ 0,\ 0). \end{gathered}$$

In our case

$$\begin{gathered} \sum_{i=1}^{3} \varepsilon_i^{*\prime}=12/5, \\ \delta^*=(0,\ -9/5,\ 0,\ 1/5,\ -4/5). \end{gathered}$$

Therefore we conclude that case (c) applies. We proceed with a successive elementary transformation of vector Y' into $Y''=Y'(\theta_0')$. We have

$$\theta_0'=\frac{175}{36},\quad \Delta''=(0,\ 11,\ 0,\ 0,\ 4),\quad Y''=(5,\ -5,\ 5).$$

The restraint vectors of the auxiliary problem $(C_{Y''})$ consist of three restraint vectors of problem (A), namely A_1, A_3, and A_4, and the unit vectors e_1, e_2, e_3. We solve problem $(C_{Y''})$ starting with program $(\Xi^{*\prime},\ E^{*\prime})$. We obtain the quasiprogram

$$X''=(2,\ 0,\ 0,\ 2,\ 0).$$

For quasiprogram X''

$$\begin{gathered} \sum \varepsilon_i^{*\prime\prime}=2, \\ \delta''^*=(0,\ -1,\ -2,\ 0,\ -1). \end{gathered}$$

The residue of program X'' is positive, and all the components of vector $\delta^{*\prime\prime}$ are nonpositive. According to § 1 this indicates that the restraints of problem (A) are inconsistent. Indeed, subtracting the second restraint of problem (A) from the third, we find

$$x_2+x_3+x_5=-2.$$

This equality cannot hold with nonnegative variables. Problem (A) is unsolvable.

§ 3. Geometrical interpretation

3-1. The Hungarian method can be interpreted geometrically in the $(m+1)$-dimensional space of points $U=(u_1, u_2, \ldots, u_m, u_{m+1})$.

Problem (A) is solved by the Hungarian method by first choosing a feasible program Y of the dual problem $(\tilde{A})$. A geometrical analog of program Y is the hyperplane Π containing the origin and passing above the cone K of problem (A). Hyperplane Π contains the augmented vectors $\overline{A}_j$ for $j \in E_Y$. The intersection of hyperplane Π and cone K is also a polyhedral cone. We denote this cone by K_Π. Polyhedral cone K_Π is generated by augmented vectors $\overline{A}_j$ for $j \in E_Y$. For $m=2$, the cone K_Π may either contain a single point (the origin), or else coincide with an edge of cone K or with a plane angle — a two-dimensional cone. K_Π is precisely m-dimensional if Y is a support program of problem $(\tilde{A})$.

To program Y of problem $(\tilde{A})$ corresponds auxiliary problem (C_Y). The cone K_Y of problem (C_Y) is defined by the restraint vectors $A_j (j \in E_Y)$ and by the augmented unit vectors

$$\overline{e}_i = \Big(\underbrace{\overbrace{0, \ldots, 0, 1}^{i}, 0, \ldots, 0}_{m}, 1\Big), \qquad i=1, 2, \ldots, m.$$

Polyhedral cone K_Y is thus tangent to hyperplane $u_{m+1}=0$ along the face containing the vectors $A_j (j \in E_Y)$. Problem (C_Y) is thus solved by determining the lowest point of intersection of line Q and cone K_Y. (In the auxiliary problem we seek the minimum of the linear form.)

Clearly, problem (C_Y) is always solvable. Solving the auxiliary problem, we find the optimal programs of the dual pair of problems (C_Y) of $(\tilde{C}_Y)$. To the solution of problem $(\tilde{C}_Y)$ corresponds hyperplane π_Y situated below the cone K_Y. Let $M^*=(\mu_1^*, \ldots, \mu_m^*, -1)$ be the direction vector of hyperplane π_Y and Q_Y the lowest point of intersection of cone K_Y and line Q. The point Q_Y is the image of the solution of problem (C_Y). The vector Q_Y can be written in terms of basis vectors of the solution of problem (C_Y) in the form

$$Q_Y = \sum_{j \in E_Y} x_j A_j + \sum_{i=1}^{m} \varepsilon_i \overline{e}_i^{\,*}.$$

Consider the point

$$P_Y = \sum_{j \in E_Y} x_j A_j \qquad \Big(P_Y = Q_Y - \sum_{i=1}^{m} \varepsilon_i \overline{e}_i^{\,*}\Big)$$

on hyperplane $u_{m+1}=0$. Through point P_Y let the line P parallel to the Ou_{m+1}-axis pass. The highest point of intersection X of line P and cone K of problem (A) is the image of quasiprogram X.

The *distance* between the points $U'=(u_1', \ldots, u_{m+1}')$ and $U''=(u_1'', \ldots, u_{m+1}'')$ is defined as

$$\sum_{i=1}^{m+1} |u_i'' - u_i'|.$$

Point X, the image of quasiprogram X, has the following extremum property: the point X is the nearest (in the sense of the preceding definition of distance) to the highest point of intersection of the cone K and the line Q of all the points $U=(u_1, \ldots, u_m, u_{m+1})$ which

(a) belong to cone $K (x_j \geqslant 0)$;

(b) belong to hyperplane Π:

$$\sum_{i=1}^{m} a_{ij} y_i \begin{cases} \geqslant c_j, & j=1, \ldots, n, \\ = c_j, & j \in E_Y; \end{cases}$$

(c) satisfy the restraints

$$u_i \leqslant b_i \quad \left(\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i\right).$$

Also in the sense of the preceding definition of distance, the length of segment $Q_Y B$ (B is the point corresponding to the constraint vector) is given by the residue

$$\varepsilon^* = \sum_{i=1}^{m} \varepsilon_i^*$$

of quasiprogram X.

Two cases are possible: point Q_Y, the image of the solution of problem (C_Y), does or does not belong to hyperplane $u_{m+1}=0$. In the former case quasiprogram X solves problem (A) (case (a)). Indeed, if point Q_Y coincides with point B, the constraint vectors belongs to the cone generated by the restraint vectors A_j $(j \in E_Y)$. Hence, the vector B is a linear combination of vectors A_j $(j \in E_Y)$ with nonnegative coefficients. A similar linear combination of the augmented vectors $\bar{A}_j$ $(j \in E_Y)$ gives the highest point of intersection of Q and the cone K of problem (A).

In this case program Y solves problem $(\tilde{A})$, and X is the optimal program of the primal problem.

To analyze the case when point Q_Y does not belong to hyperplane $u_{m+1}=0$, we consider the geometrical interpretation of the elementary transformation of program Y of problem $(\tilde{A})$ into program Y'.

Let $Y^* = (y_1, y_2, \ldots, y_m, -1)$ be the direction vector of hyperplane Π. The direction vector of hyperplane π_Y corresponding to the solution of problem $(\tilde{C}_Y)$, was denoted by $M^* = (\mu_1^*, \ldots, \mu_m^*, -1)$.

Consider hyperplane $\bar{\pi}_Y$ which is the intersection of hyperplanes π_Y and $u_{m+1}=0$ parallel to the Ou_{m+1}-axis. The first m components of the direction vectors $\bar{M}^*$ and M^* of hyperplanes $\bar{\pi}_Y$ and π_Y are the same. The last component of vector $\bar{M}^*$ is zero.

The vector

$$Y^*(\theta) = Y^* - \theta \bar{M}^*$$

is the direction vector of hyperplane $\Pi(\theta)$. For $\theta=0$ hyperplane $\Pi(\theta)$ coincides with Π. As θ increases, hyperplane $\Pi(\theta)$ rotates about the intersection S of hyperplanes Π and $\bar{\pi}_Y$.

The hyperplane rotates (θ increases) as long as $\Pi(\theta)$ does not capture one of the augmented vectors $\bar{A}_j$ $(j \notin E_Y)$. Let this occur for $\theta=\theta_0$. Hyperplane $\Pi' = \Pi(\theta_0)$ corresponds to a new program Y' of the dual problem $(\tilde{A})$. This follows from the fact that none of the vectors $\bar{A}_j$ $(j \in E_Y)$ can be above hyperplane Π', since

$$(Y'(\theta), A_j) = 0 - \theta\delta_j^* \geqslant 0.$$

Observe that the intersection of hyperplane π_Y with the coordinate hyperplane $u_{m+1}=0$ contains all the basis vectors A_j $(j \in E_Y)$ of the solution of problem (C_Y). The corresponding augmented vectors $\bar{A}_j$ belong to hyperplane S, the intersection of Π and $\bar{\pi}_Y$, and, consequently, to hyperplane $\Pi' = \Pi(\theta_0)$.

Hyperplane π_Y passes below the cone K_Y. Hence, all the vectors $\overline{A}_j$ $(j \in E_Y)$ lie on one side of hyperplane $\overline{\pi}_Y$. The other restraint vectors $\overline{A}_j$ $(j \notin E_Y)$ need not meet this condition. They fall on either side of $\overline{\pi}_Y$.

We shall consider two cases.

First let all $\overline{A}_j$ $(j \notin E_Y)$ lie in the same halfspace relative to $\overline{\pi}_Y$ as the vectors $\overline{A}_j (j \in E_Y)$. In this case hyperplane $\overline{\pi}_Y$ separates the cone K and the line Q. Indeed, under our assumptions,

$$(\overline{A}_j, \overline{M}^*) = \delta_j^* \leqslant 0, \qquad j = 1, 2, \ldots, n. \tag{3.1}$$

On the other hand

$$(B + \lambda e_{m+1}, \overline{M}^*) = (B, \overline{M}^*) = \sum_{i=1}^{m} b_i \mu_i^* = \varepsilon^* > 0. \tag{3.2}$$

Vector $B + \lambda e_{m+1}$ defines the line Q. The first and the second equalities in (3.2) are obvious. The third equality follows from the duality theorem for the auxiliary problem.

Inequalities (3.1) and (3.2) indicate that line Q and cone K do not intersect. In this case, $\theta_* = \infty$, i. e., none of the vectors A_j $(j \notin E_Y)$ is captured by hyperplane $\Pi(\theta)$ for any finite value $\theta \geqslant 0$.

Thus, if all the vectors $\overline{A}_j$ $(j \notin E_Y)$ fall on the same side of hyperplane $\overline{\pi}_Y$ as the vectors $\overline{A}_j$ $(j \in E_Y)$, the restraints of problem (A) are inconsistent (case (b)).

If at least one of the vectors $\overline{A}_j (j \notin E_Y)$ and the vectors $\overline{A}_j$ $(j \in E_Y)$ lie on different sides of $\overline{\pi}_Y$, we have case (c). Hyperplane Π is rotated about S (the intersection of Π and $\overline{\pi}_Y$) until some $\overline{A}_j$ $(j \notin E_Y)$ is captured. The corresponding hyperplane Π' gives a new program Y' of the dual problem, which generates an auxiliary problem and a quasiprogram X' with a lower residue. After a finite number of steps we obtain case (a) or case (b).

3-2. We shall give the geometrical constructions illustrating the application of the Hungarian method to the determination of the optimal program of the problem solved in Chapter 4, 4-3 by the simplex method and in Chapter 6, 2-8 by the dual simplex method.

In this problem we are required to minimize the linear form

$$L(X) = 10x_1 + 8x_2 + 7x_3 + 16x_4 + 21x_5$$

subject to the following conditions

$$\begin{gathered} 4x_1 + 2x_2 + 5x_3 + 10x_4 + 5x_5 = 6, \\ 9x_1 + 10x_2 + 12.5x_3 + 18x_4 + 16.5x_5 = 14, \\ x_j \geqslant 0, \quad j = 1, 2, \ldots, 5. \end{gathered}$$

Figure 7-1 shows the restraint vectors A_j and the constraint vector B, the corresponding augmented vectors of the problem and the line Q passing through the point B parallel to the Ou_3-axis. The figure, moreover, shows the polyhedral cone K generated by the vectors $\overline{A}_j$ and the augmented unit vectors $\overline{e}_i$ essential for the geometrical interpretation of the auxiliary problem.

It is easily seen that the vector $Y = (6, -2/5)$ is a feasible program of the auxiliary problem in which we try to minimize the linear form

$$\tilde{L}(Y) = 6y_1 + 14y_2$$

subject to the conditions

$$\begin{aligned} 4y_1 + 9y_2 &\geqslant 10, \\ 2y_1 + 10y_2 &\geqslant 8, \\ 5y_1 + 12.5y_2 &\geqslant 7, \\ 10y_1 + 18y_2 &\geqslant 16, \\ 5y_1 + 16.5y_2 &\geqslant 21. \end{aligned}$$

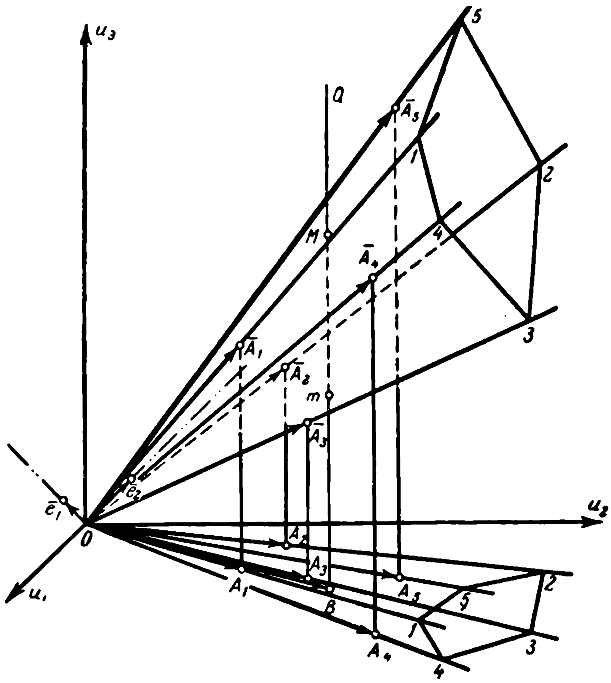


FIGURE 7.1

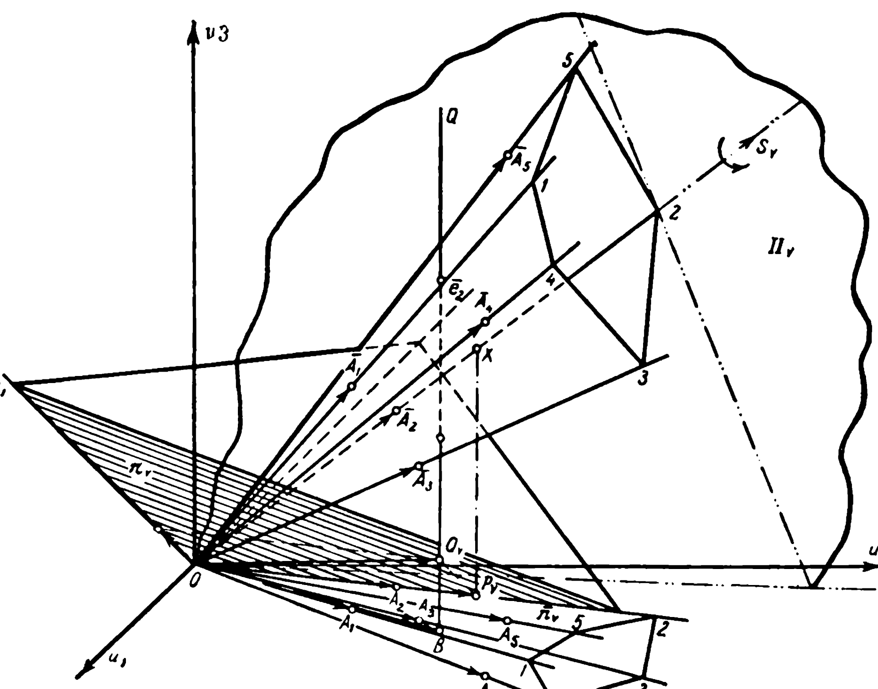


FIGURE 7.2

Program Y reduces the second restraint to an equality. The set E_Y comprises in our case only one index, $j=2$. Hence, the plane Π_Y, the geometrical image of program Y, contains the vector $\bar{A}_2$. The equation of plane Π_Y has the form

$$6u_1-\frac{2}{5}u_2=u_3.$$

The plane Π_Y extends above the cone K. Program Y generates auxiliary problem (C_Y). The restraint vectors of problem (C_Y) are A_2, $\bar{e}_1$, $\bar{e}_2$. The polyhedral cone K_Y corresponding to the auxiliary problem is shown in Figure 7.2. This figure also shows the cone K of the primal problem and the plane Π_Y — the image of program Y of the auxiliary problem. The line Q intersects the cone K_Y at two points. The lowest point of intersection, Q_Y, of line Q and cone K_Y lie in the plane Π_Y spanned by the vectors $\bar{e}_1$ and A_2. The plane Π_Y is the geometrical image of the optimal program of problem $(\tilde{C}_Y)$, the dual of problem (C_Y). Point Q_Y corresponds to the solution of problem (C_Y). We resolve the vector Q_Y along the edges $\bar{e}_1$, $\bar{e}_2$, A_2 of cone K_Y. Point P_Y is the projection of point Q_Y on vector A_2. The line drawn from point P_Y parallel to Ou_3 intersects the cone K of the primal problem at two points. The highest of these points, point X, is the geometrical image of quasiprogram X specified by the solution of problem (C_Y).

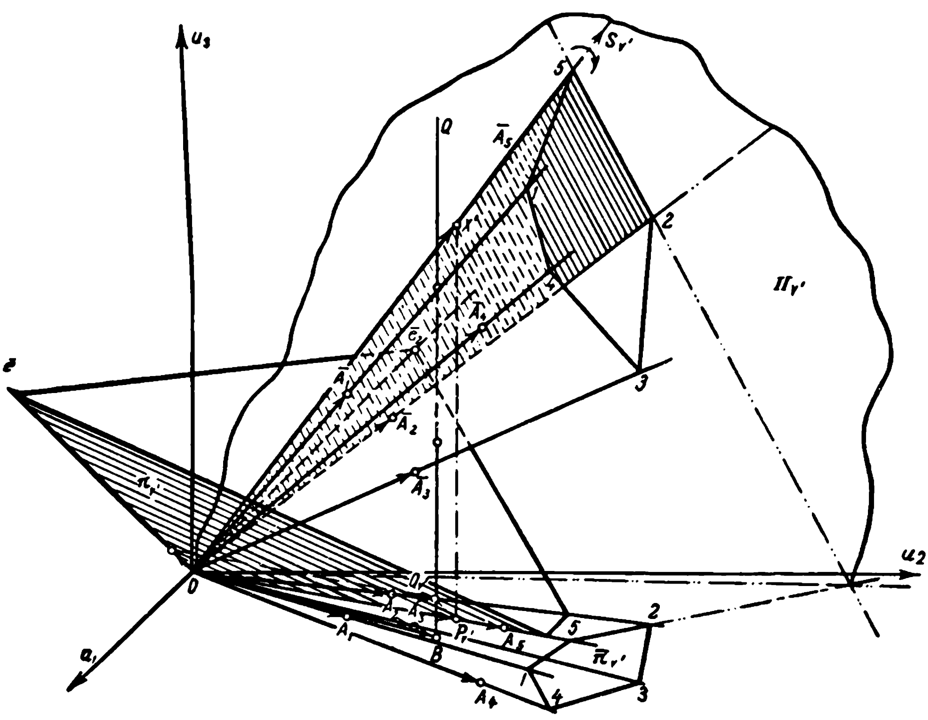


FIGURE 7.3

The length of segment Q_YB (in the sense of the previous definition of distance), i. e., the minimum value of the linear form of the auxiliary problem (C_Y), is equal to ε^*— the residue of quasiprogram X. Since $\varepsilon^*=16/5\neq0$, quasiprogram X is not the solution of the problem. The length of segment XP_Y gives the value of the linear form of the primal problem on quasiprogram X, $L(X)=56/5$. The plane π_Y intersects the coordinate plane $u_3=0$ in the direction of vector A_2. The plane $\bar{\pi}_Y$ is, thus, defined by the vector A_2 and the Ou_3-axis or, equivalently, by vectors A_2 and $\bar{A}_2$.

The planes Π_Y and $\bar{\pi}_Y$ intersect along the line S_Y parallel to vector $\bar{A}_2$. We rotate the plane Π_Y about the line S_Y in the sense of the arrow until one of the augmented vectors $\bar{A}_j$ is captured. The first vector

to be captured in the process is $\overline{A}_5$. The corresponding position of the plane will be denoted by $\Pi_{Y'}$. The plane $\Pi_{Y'}$ spanned by vectors $\overline{A}_2$ and $\overline{A}_5$ corresponds to a new program Y' of the dual problem $(\tilde{A})$. The vector Y' is a support program of problem $(\tilde{A})$. Program Y' defines a new auxiliary program. Geometrically this means that the plane $\Pi_{Y'}$ defines the cone $K_{Y'}$ of problem $(C_{Y'})$. The tetrahedral cone $K_{Y'}$ is generated by the augmented unit vectors $\overline{e}_1$, $\overline{e}_2$ and the restraint vectors A_2 and A_5.

The constructions of the second iteration are shown in Figure 7.3. As in the first iteration, we find the point $Q_{Y'}$ corresponding to the solution of the auxiliary problem $(C_{Y'})$. The point $Q_{Y'}$ lies above the plane $u_3=0$. Hence, case (a) does not apply. The line of intersection of plane $\pi_{Y'}$ with plane $u_3=0$ extends along the vector A_5. The direction of the vector A_5 and the Ou_3-axis define the plane $\overline{\pi}_{Y'}$. The restraint vectors $\overline{A}_j$ $(j \notin E_{Y'} = \{5\})$ lie on either side of plane $\overline{\pi}_{Y'}$. Hence, case (b) does not apply either.

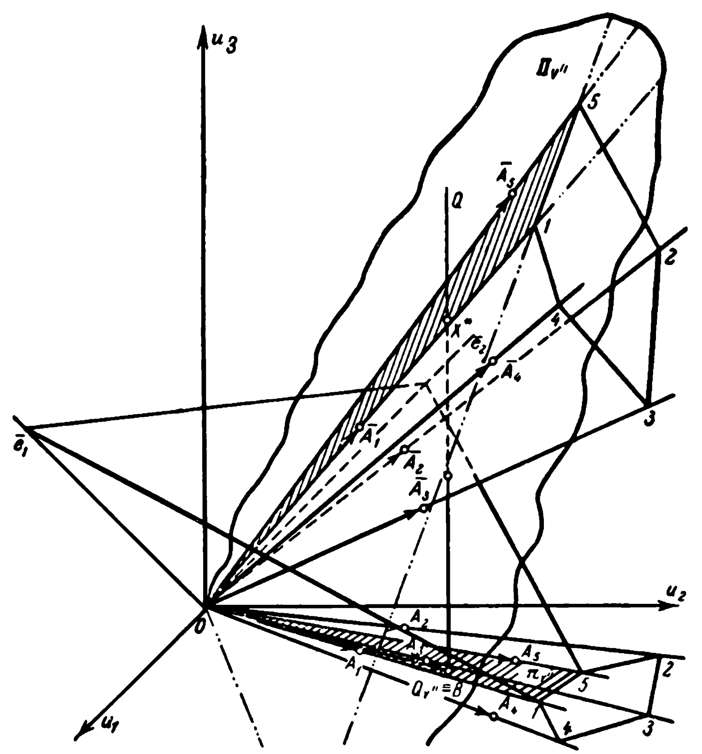


FIGURE 7.4

Following the same rules as in the first iteration, we find the point X', the image of quasiprogram X' determined by program Y' of the dual problem. In our case $S_{Y'}$, the line of intersection of planes $\Pi_{Y'}$ and $\overline{\pi}_{Y'}$, extends along the vector $\overline{A}_5$.

We rotate the plane $\Pi_{Y'}$ about the line $S_{Y'}$, the line of intersection of planes $\Pi_{Y'}$ and $\overline{\pi}_{Y'}$, until another vector $\overline{A}_j$ $(j \notin E_{Y'})$ is captured. We thus obtain a plane $\Pi_{Y''}$ corresponding to program Y'' of the dual problem $(\tilde{A})$. The plane $\Pi_{Y''}$ is spanned by vectors $\overline{A}_5$ and $\overline{A}_1$. The vector Y'' is a support program of problem $(\tilde{A})$. The cone $K_{Y''}$ of the auxiliary problem $(C_{Y''})$ is generated by vectors $\overline{e}_1$, $\overline{e}_2$, A_1, and A_5.

Point $Q_{Y''}$, the image of the solution of auxiliary problem $(C_{Y''})$, coincides with point B. The residue of quasiprogram X'' is zero and quasiprogram X'' is thus the solution of the primal problem. The constructions connected with the last iteration are shown in Figure 7.4.

§ 4. The case of bilateral restraints

4-1. The Hungarian method, like other linear-programming methods discussed in previous chapters, can be extended to problems with bilateral restraints. How this is done will be shown below.

Consider a general problem with bilateral restraints: maximize the linear form

$$L(X)=\sum_{j=1}^{n} c_j x_j \tag{4.1}$$

subject to the conditions

$$\sum_{j=1}^{n} A_j x_j = B, \tag{4.2}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j, \qquad j=1, 2, \ldots, n. \tag{4.3}$$

Here

$$A_j=(a_{1j}, a_{2j}, \ldots, a_{mj})^T, \quad B=(b_1, b_2, \ldots, b_m)^T.$$

The rank of matrix $(A_1, A_2, \ldots, A_n)$ is assumed to be m.

Each variable x_j of problem (4.1)-(4.3) can be restrained on two sides or on one side only. In the latter case, the second bound is assumed to be at infinity ($\alpha_j=-\infty$ or $\beta_j=\infty$). Primal problem (4.1)—(4.3) will be denoted as problem (A). In the following we shall deal with kernels of programs of problem $(\tilde{A})$, the dual of problem (A). Without repeating the statement of problem $(\tilde{A})$, which is given in Chapter 6, 3-2, we write only the conditions satisfied by a program kernel of this problem.

The vector $Y=(y_1, y_2, \ldots, y_m)$ is the kernel of a feasible program of problem $(\tilde{A})$ if and only if

$$\Delta_j=\begin{cases} \sum_{i=1}^{m} a_{ij} y_i - c_i \geqslant 0, & \text{if} \quad \beta_j=\infty, \\ \sum_{i=1}^{m} a_{ij} y_i - c_j \leqslant 0, & \text{if} \quad \alpha_j=-\infty. \end{cases} \tag{4.4}$$

It follows from (4.4) that when all the problem variables are restrained on two sides ($\alpha_j \neq -\infty$, $\beta_j \neq \infty$), any vector Y is the kernel of some feasible program of problem $(\tilde{A})$. To each kernel Y there correspond, generally speaking, several feasible programs of problem $(\tilde{A})$. If we choose from these programs that which minimizes the linear form of problem $(\tilde{A})$, we establish a one-to-one correspondence between the kernels and the programs of problem $(\tilde{A})$.

Let $\tilde{L}(Y)$ be the value of the linear form of problem $(\tilde{A})$ on the program corresponding to kernel Y. We have

$$\tilde{L}(Y)=\sum_{i=1}^{m} b_i y_i - \sum_{j=1}^{n} \Delta_j \gamma_j, \tag{4.5}$$

where

$$\gamma_j=\begin{cases} \alpha_j, & \text{if} \quad \Delta_j>0, \\ \beta_j, & \text{if} \quad \Delta_j<0. \end{cases} \tag{4.6}$$

Here, as usual,

$$\Delta_j=\sum_{i=1}^{m} a_{ij} y_i - c_j. \tag{4.7}$$

Conditions (4.4) and equality (4.5) are derived in Chapter 6, 3-2.

4-2. We now extend the concepts of auxiliary and augmented problems, introduced in § 1, to problems with bilateral restraints.

The augmented problem is to minimize the linear form

$$\sum_{i=1}^{m} \varepsilon_i \tag{4.8}$$

subject to the conditions

$$\sum_{j=1}^{n} a_{ij}x_j + \varepsilon_i = b_i, \qquad i = 1, 2, \ldots, m; \tag{4.9}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j, \quad \varepsilon_i \geqslant 0; \tag{4.10}$$
$$j = 1, 2, \ldots, n; \quad i = 1, 2, \ldots, m.$$

This problem will be denoted as problem (B). Let $Y = (y_1, y_2, \ldots, y_m)$ be the kernel of a feasible program of problem $(\tilde{A})$. Let E_Y be the set of indices j such that

$$\Delta_j = \sum_{i=1}^{m} a_{ij}y_i - c_j = 0.$$

Each kernel Y is associated with an auxiliary problem (problem (C_Y)), which is formed from problem (B) by adding restraints: $x_j = \gamma_j$, $j \notin E_Y$, where γ_j are defined in (4.6).

Problem (C_Y) thus involves minimizing the linear form

$$\sum_{i=1}^{m} \varepsilon_i \tag{4.8}$$

subject to the conditions

$$\sum_{j \in E_Y} a_{ij}x_j + \varepsilon_i = b_i^{(Y)}; \tag{4.11}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j; \quad \varepsilon_i \geqslant 0; \tag{4.12}$$
$$j \in E_Y, \quad i = 1, 2, \ldots, m.$$

Here

$$b_i^{(Y)} = b_i - \sum_{j \notin E_Y} \gamma_j a_{ij}. \tag{4.13}$$

The restraint vectors of the auxiliary problem (C_Y) are some of the restraint vectors A_j of problem (A) and the unit vectors $e_1, e_2, \ldots, e_m$. The constraint vector

$$B^{(Y)} = (b_1^{(Y)}, b_2^{(Y)}, \ldots, b_m^{(Y)})$$

of problem (C_Y) is formed from the constraint vector B of problem (A) according to (4.13).

Let problem $(\tilde{C}_Y)$ be the dual problem of (C_Y). Every program (Ξ, E),

$$\Xi = (\xi_j), \quad j \in E_Y; \quad E = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m),$$

of problem (C_Y) can be made to correspond to vector $X = (x_1, x_2, \ldots, x_n)$ with the coordinates

$$x_j = \begin{cases} \xi_j, & j \in E_Y, \\ \gamma_j, & j \notin E_Y, \end{cases} \tag{4.14}$$

where γ_j are defined in (4.6).

Clearly, the vector X satisfies restraints (4.3), and

$$b_i - \sum_{j=1}^{n} a_{ij}x_j = \varepsilon_i \geqslant 0. \tag{4.15}$$

We now introduce some definitions.

The vector $X=(x_1, x_2, \ldots, x_n)$ corresponding to the support program (Ξ^*, E^*) of any auxiliary problem (C_Y) will be called a **quasiprogram** of the primal problem (A). Any quasiprogram satisfies restraints (4.3) of problem (A). According to (4.15), the parameter $\varepsilon_i^* \geqslant 0$ specifies the residue of the i-th restraint of system (4.2) when quasiprogram X is substituted in this restraint. The sum $\varepsilon_0^* = \sum_{i=1}^{m} \varepsilon_i^*$ will be called the **residue** of quasiprogram X.

Since $\varepsilon_i^* \geqslant 0$, the condition $\varepsilon_0^* = 0$ implies that all ε_i^* are zero.

A quasiprogram with zero residue thus satisfies equalities (4.2) and is, consequently, a feasible program of problem (A).

4-3. The process of solution of problem (A) by the Hungarian method involves transferring quasiprograms of the problem. We shall now give a condition sufficient for the process of solution to yield the optimal program of the problem:

Optimality test. *If the residue ε_0^* of quasiprogram X vanishes, X is the optimal program of the primal problem.*

Proof. As we have already seen any quasiprogram of problem (A) with a zero residue is a feasible program of the problem. The vector X is, thus, a feasible program of problem (A). Let program X correspond to the solution of problem (C_Y) where $Y=(y_1, y_2, \ldots, y_m)$ is the kernel of a feasible program of problem $(\tilde{A})$. Since the program X is formed from the solution of problem (C_Y) according to (4.14), we have

$$\begin{aligned} \Delta_j = \sum_{i=1}^{m} a_{ij} y_i - c_j = 0, &\quad \text{if} \quad \alpha_j < x_j < \beta_j, \\ \Delta_j \geqslant 0, &\quad \text{if} \quad x_j = \alpha_j, \\ \Delta_j \leqslant 0, &\quad \text{if} \quad x_j = \beta_j. \end{aligned}$$

According to the second form of the optimality test established in Chapter 4, § 5, these conditions show optimality of program X. This completes the proof.

Let $Y=(y_1, y_2, \ldots, y_m)$ be the kernel of a feasible program of problem $(\tilde{A})$. Consider auxiliary problem (C_Y) generated by this kernel. Problem (C_Y) can be solved by the simplex method described for problems with bilateral restraints in Chapter 4, § 5. Let the set of numbers ξ_j^* $(j \in E_Y)$, ε_i^*, $i=1, 2, \ldots, m$ define a support solution of the auxiliary problem obtained by the simplex method. Then, according to the second form of the optimality test (Chapter 4, 5-3), there exists a vector $M^* = (\mu_1^*, \mu_2^*, \ldots, \mu_m^*)$ such that

$$\sum_{i=1}^{m} a_{ij} \mu_i^* \begin{cases} =0, & \text{if} \quad \alpha_j < \xi_j^* < \beta_j, \\ \leqslant 0, & \text{if} \quad \xi_j = \alpha_j, \\ \geqslant 0, & \text{if} \quad \xi_j = \beta_j, \end{cases} \tag{4.16}$$

$$\mu_i^* \begin{cases} =1, & \text{if} \quad \varepsilon_i^* > 0, \\ \leqslant 1, & \text{if} \quad \varepsilon_i^* = 0. \end{cases}$$

If we use the second simplex algorithm, the vector M^* is determined at the same time as the solution of problem (C_Y). We leave it to the reader (see Exercise 6) to show that M^* is the kernel of the solution of problem $(\tilde{C}_Y)$.

Let $X=(x_1, x_2, \ldots, x_n)$ be a quasiprogram of problem (A) corresponding to the given solution of problem (C_Y).

From (4.14) we have

$$x_j = \begin{cases} \xi_j^*, & j \in E_Y, \\ \gamma_j, & j \notin E_Y, \end{cases} \tag{4.17}$$

where γ_j, equal to α_j or β_j, are defined in (4.6). Let

$$\delta_j^* = \sum_{i=1}^{m} a_{ij}\mu_i^*, \qquad j = 1, 2, \ldots, n. \tag{4.18}$$

Three cases arise in the analysis of quasiprogram X depending on the value of its residue $\varepsilon_0^* = \sum_{i=1}^{m} \varepsilon_i^*$ and the relative signs of Δ_j and δ_j^* (see (4.7) and (4.18)).

(a) The residue ε_0^* of quasiprogram X is zero.

(b) The residue $\varepsilon_0^* > 0$, and

$$\delta_j^* \begin{cases} \geqslant 0, & \text{if } \Delta_j < 0, \\ \leqslant 0, & \text{if } \Delta_j > 0. \end{cases} \tag{4.19}$$

(c) The residue $\varepsilon_0^* > 0$, but conditions (4.19) do not hold.

We shall consider each of these cases separately.

If $\varepsilon_0^* = 0$ (case (a)), then, according to the optimality test, quasiprogram X is the optimal program of problem (A). The other two cases are conveniently investigated using the elementary transformation introduced in § 1.

4-4. An elementary transformation of kernel Y is defined as

$$Y(\theta) = Y - \theta M^*, \tag{4.20}$$

or, equivalently,

$$y_i(\theta) = y_i - \theta\mu_i^*, \qquad i = 1, 2, \ldots, m. \tag{4.20'}$$

Let

$$\Delta_j(\theta) = \sum_{i=1}^{m} a_{ij} y_i(\theta) - c_j, \qquad j = 1, 2, \ldots, n.$$

Applying equalities (4.20') and the definition of Δ_j and δ_j^*, we have

$$\Delta_j(\theta) = \sum_{i=1}^{m} a_{ij} y_i - \theta \sum_{i=1}^{m} a_{ij}\mu_i^* - c_j = \Delta_j - \theta\delta_j^*, \quad j = 1, 2, \ldots, n. \tag{4.21}$$

According to (4.16) and (4.18), we obtain for $j \in E_Y$:

$$\delta_j^* \begin{cases} = 0 & \text{for } \alpha_j < x_j < \beta_j, \\ \leqslant 0 & \text{for } \alpha_j = x_j, \\ \geqslant 0 & \text{for } x_j = \beta_j. \end{cases}$$

Hence, for any $\theta \geqslant 0$ and $j \in E_Y$, we have

$$\Delta_j(\theta) = \Delta_j - \theta\delta_j^* = -\theta\delta_j^* \begin{cases} = 0, & \text{if } \alpha_j < x_j < \beta_j, \\ \geqslant 0, & \text{if } \alpha_j = x_j, \\ \leqslant 0, & \text{if } x_j = \beta_j. \end{cases} \tag{4.22}$$

From the set of all elementary transformations (4.20) we choose those whose parameter $\theta \geqslant 0$ satisfies

$$\Delta_j \Delta_j(\theta) \geqslant 0, \; j \notin E_Y. \tag{4.23}$$

In other words, we take a set of nonnegative θ on which Δ_j and $\Delta_j(\theta)$ have the same sign for all $j \notin E_Y$. If $\Delta_j > 0$, the j-th restraint (4.23) may be violated

only if $\delta_j^* > 0$, which occurs if

$$\theta = \frac{\Delta_j}{\delta_j^*} > 0. \tag{4.24}$$

Analogously, if $\Delta_j < 0$ the corresponding restraint in (4.23) is not satisfied only for $\delta_j^* < 0$. The value of θ on which the j-th restraint is not satisfied in this case is also given by (4.24). Hence, all the inequalities in system (4.24) are satisfied if and only if

$$\theta \leqslant \theta_0 = \min_{\frac{\Delta_j}{\delta_j^*} > 0} \frac{\Delta_j}{\delta_j^*}, \tag{4.25}$$

where the minimum is taken over the indices j such that

$$\frac{\Delta_j}{\delta_j^*} > 0, \quad \delta_j^* \neq 0.$$

If for any $j \notin E_Y$, $\frac{\Delta_j}{\delta_j^*} < 0$ or $\delta_j^* = 0$, then, by definition, $\theta_0 = \infty$, i.e., the parameter θ is not bounded above.

In the following we shall confine the discussion to elementary transformations with nonnegative values of the parameter θ satisfying condition (4.25).

Note that according to (4.22) and (4.23), with $\alpha_j = -\infty$ $(\beta_j = \infty)$

$$\Delta_j(\theta) \leqslant 0 \quad (\Delta_j(\theta) \geqslant 0).$$

Therefore, with the specified values of θ, the vector $Y(\theta)$ is the kernel of a feasible program of problem $(\tilde{A})$ (conditions (4.4) are satisfied).

Now let us note the change in the linear form of problem $(\tilde{A})$ under elementary transformation (4.20) with a parameter $(\theta \geqslant 0)$ satisfying inequality (4.25). According to (4.5), the value of the linear form of problem $(\tilde{A})$ on the program corresponding to kernel $Y(\theta)$ is given by

$$\tilde{L}(Y(\theta)) = \sum_{i=1}^{m} b_i y_i(\theta) - \sum_{j=1}^{n} \gamma_j(\theta)\, \Delta_j(\theta), \tag{4.26}$$

where

$$\gamma_j(\theta) = \begin{cases} \alpha_j, & \text{if } \Delta_j(\theta) > 0, \\ \beta_j, & \text{if } \Delta_j(\theta) < 0. \end{cases}$$

We now transform equality (4.26). If $j \in E_Y$ and $\Delta_j(\theta) \neq 0$, then, from (4.22),

$$\gamma_j(\theta) = x_j.$$

If now $j \notin E_Y$ then, by assumption, $\Delta_j(\theta)\, \Delta_j \geqslant 0$. Hence, from the definition (4.17) of quasiprogram X, we again have $\gamma_j(\theta) = x_j$.

We may thus take

$$\gamma_j(\theta) = x_j \quad \text{for} \quad j = 1, 2, \ldots, n. \tag{4.27}$$

Applying (4.27), and also (4.20') and (4.21), we obtain

$$\tilde{L}(Y(\theta)) = \sum_{i=1}^{m} b_i y_i - \sum_{j=1}^{n} x_j \Delta_j - \theta \left[\sum_{i=1}^{m} b_i \mu_i^* - \sum_{j=1}^{n} x_j \delta_j^*\right]. \tag{4.28}$$

Since $Y = Y(0)$, we have

$$\sum_{i=1}^{m} b_i y_i - \sum_{j=1}^{n} x_j \Delta_j = \tilde{L}(Y). \tag{4.29}$$

The value L_C of the linear form of problem (C_Y) on the program defined by the set of numbers $\xi_j^* = x_j$, $j \in E_Y$, ε_i^*, $i = 1, 2, \ldots, m$ is given by

$$L_C = \varepsilon_0^* = \sum_{i=1}^{m} \varepsilon_i^*. \tag{4.30}$$

The value $\tilde{L}_C$ of the linear form of problem $(\tilde{C}_Y)$ on the program corresponding to kernel M^* is given by

$$\tilde{L}_C = \sum_{i=1}^{m} b_i^{(Y)} \mu_i^* - \sum_{j \in E_Y} x_j \delta_j^*. \tag{4.31}$$

Here we used the statement of the auxiliary problem (4.8), (4.11), (4.12), equality (4.5), and also (4.22).

In deriving (4.5), maximization of the linear form of the primal problem was required, It is easily seen (see Exercise 7) that (4.5) also applies when minimization of the linear form of the primal problem is required. In this case, however, γ_j are defined as

$$\gamma_j = \begin{cases} \alpha_j, & \text{if} \quad \Delta_j < 0, \\ \beta_j, & \text{if} \quad \Delta_j > 0. \end{cases}$$

Relationship (4.22) shows that in this case $\gamma_j = x_j$. We transform (4.31) applying (4.13) for $b_i^{(Y)}$:

$$\tilde{L}_C = \sum_{i=1}^{m} b_i \mu_i^* - \sum_{i=1}^{m} \Big(\sum_{j \notin E_Y} a_{ij} x_j\Big) \mu_i^* - \sum_{j \in E_Y} x_j \delta_j^* = \sum_{i=1}^{m} b_i \mu_i^* - \sum_{j \notin E_Y} x_j \sum_{i=1}^{m} a_{ij} \mu_i^* - \sum_{j \in E_Y} x_j \delta_j^*.$$

Further applying (4.18), we obtain

$$\tilde{L}_C = \sum_{i=1}^{m} b_i \mu_i^* - \sum_{j=1}^{n} x_j \delta_j^*. \tag{4.32}$$

Since L_C and $\tilde{L}_C$ are the optimal values of the linear forms of problems (C_Y) and $(\tilde{C}_Y)$, respectively, we have from the first duality theorem (Chapter 3, Theorem 4.1),

$$L_C = \tilde{L}_C.$$

Hence, applying (4.30) and (4.32), we obtain

$$\varepsilon_0^* = \sum_{i=1}^{m} \varepsilon_i^* = \sum_{i=1}^{m} b_i \mu_i^* - \sum_{j=1}^{n} x_j \delta_j^*. \tag{4.33}$$

Using (4.29) and (4.33), we rewrite (4.28) as

$$\tilde{L}(Y(\theta)) = \tilde{L}(Y) - \theta \varepsilon_0^*, \tag{4.34}$$

where ε_0^* is the residue of quasiprogram X corresponding to the solution of the auxiliary problem (C_Y).

We emphasize that equality (4.34) was derived under the assumption that $0 \leqslant \theta \leqslant \theta_0$.

4-5. We are now in a position to give a comprehensive analysis of cases (b) and (c) stated in 4-3.

Consider case (b). According to (4.19), $\delta_j^* \neq 0$ and Δ_j have different signs for all $j \notin E_Y$. Hence, in this case $\theta_* = \infty$. Therefore, the vector $Y(\theta)$ is the kernel of a program of problem $(\tilde{A})$ for any $\theta \geqslant 0$. Further, since the residue ε_0^* of quasiprogram X is positive and equality (4.34) applies for all $\theta \geqslant 0$, we conclude that the linear form of problem $(\tilde{A})$ is unbounded in the set of its feasible programs. Hence, also, the primal problem (A) is unsolvable

because its restraints are inconsistent. Case (b) thus indicates unsolvability of primal problem (A).

Case (b) is sometimes called unsolvability test of the primal problem.

In case (c) the residue ε_0^* of quasiprogram X is positive and for some $j \notin E_Y$

$$\frac{\Delta_j}{\delta_j^*}>0, \quad \delta_j^* \neq 0.$$

From (4.25) we have

$$0<\theta_0=\frac{\Delta_{j_0}}{\delta_{j_0}^*}<\infty, \tag{4.35}$$

where j_0 is one of the indices j on which θ_0 is obtained. Let $Y'=Y(\theta_0)$. We proved in the previous article that the vector Y' is the kernel of a program of problem $(\tilde{A})$, and according to (4.33)

$$\tilde{L}(Y')=\tilde{L}(Y)-\theta_0\varepsilon_0^*.$$

Hence, since θ_0 and ε_0^* are both positive, we obtain

$$\tilde{L}(Y')<\tilde{L}(Y).$$

The transition to kernel Y' thus decreases the linear form of problem $(\tilde{A})$.

The kernel Y' generates a new auxiliary problem $(C_{Y'})$ which is solved to obtain a new quasiprogram X' with a residue $\varepsilon_0^{*'}$.

We shall show that the set of numbers x_j, $j \in E_{Y'}$, ε_i^*, $i=1, 2, \ldots, m$ constitutes a support program of problem $(C_{Y'})$. Let $j \notin E_Y$, i.e., $\Delta_j'=\Delta_j(\theta_0) \neq 0$. From (4.22) (for $j \in E_Y$) and (4.23) (for $j \notin E_Y$), we find

$$x_j=\begin{cases} \alpha_j, & \text{if } \Delta_j'>0, \\ \beta_j, & \text{if } \Delta_j'<0. \end{cases}$$

Hence

$$b_i^{(Y')}=b_i-\sum_{j \notin E_{Y'}} \gamma_j a_{ij}=b_i-\sum_{j \notin E_{Y'}} x_j a_{ij}.$$

The set of numbers x_j, $j \in E_{Y'}$, ε_i^*, $i=1, 2, \ldots, m$ therefore satisfies not only conditions (4.12), but also conditions (4.11) and is, thus, a feasible program of problem $(C_{Y'})$ (in restraints (4.11) and (4.12) the subscript Y is replaced by Y').

If the restraint vector A_j of the problem (A) is one of the basis vectors of the optimal support program x_j, $j \in E_Y$, ε_i^*, $i=1, 2, \ldots, m$ of problem (C_Y), then $\delta_j^*=0$. Therefore,

$$\Delta_j'=\Delta_j-\theta_0\delta_j^*=0,$$

i.e., $j \in E_{Y'}$. Hence, the program x_j, $j \in E_{Y'}$, ε_i^*, $i=1, \ldots m$ of problem $(C_{Y'})$ is a support program. Its basis coincides with the basis of the support solution of problem (C_Y). This support program is generally fairly close to the optimal program of auxiliary problem $(C_{Y'})$. Taking it as the first program, we substantially reduce the number of iterations required to solve problem $(C_{Y'})$.

When solving a nondegenerate problem $(C_{Y'})$, the transition to quasiprogram X' lowers the residue:

$$\varepsilon_0^{*'}<\varepsilon_0^*.$$

Indeed, from (4.35),

$$\Delta'_{j_0}=\Delta_{j_0}-\theta_0\delta^*_{j_0}=0.$$

Therefore $j_0\in E_{Y'}$, i.e., A_{j_0} is a restraint vector of problem $(C_{Y'})$. Again, from (4.35)

$$\Delta_{j_0}\delta^*_{j_0}>0. \tag{4.36}$$

From the definition of quasiprogram X it follows that

$$x_{j_0}=\begin{cases}\alpha_{j_0}, & \text{if } \Delta_{j_0}>0,\\ \beta_{j_0}, & \text{if } \Delta_{j_0}<0.\end{cases}$$

Applying (4.36), we can rewrite the last equality as

$$x_{j_0}=\begin{cases}\alpha_{j_0}, & \text{if } \delta^*_{j_0}>0,\\ \beta_{j_0}, & \text{if } \delta^*_{j_0}<0.\end{cases} \tag{4.37}$$

Relationship (4.37) shows that the support program x_j, $j\in E_{Y'}$, ε^*_i, $i=1, 2, \ldots, m$ of problem $(C_{Y'})$ fulfills the requirements of case (c) of the simplex method (see Chapter 4, 5-5), and the vector A_{j_0} can be introduced into the basis. To verify this proposition, it suffices to see that problem $(C_{Y'})$ involves minimization of the linear form and case (b) cannot arise in the process of solution:

$$\sum_{i=1}^{m}\varepsilon_i\geqslant 0.$$

Since the vector A_{j_0} is introduced into the basis, we obtain a new support program of problem $(C_{Y'})$, on which the linear form $\sum_{i=1}^{m}\varepsilon_i$ assumes a lower value than

$$\varepsilon^*_0=\sum_{i=1}^{m}\varepsilon^*_i$$

(by assumption, problem $(C_{Y'})$ is nondegenerate). Hence, the minimum value $\varepsilon^{*\prime}_0$ of the linear form of problem $(C_{Y'})$ is a fortiori less than the residue ε^*_0 of program X.

In case (c) we can, thus, pass to a new quasiprogram X' with a lower residue (this applies to a nondegenerate problem $(C_{Y'})$).

4-6. The solution of a problem with bilateral restraints by the Hungarian method involves transferring its quasiprograms.

The process of solution is made up of identical iterations of which we shall describe one of these, say, the first.

Let Y be the kernel of a feasible program of problem $(\tilde{A})$. As we have already observed, if all the variables of problem (A) are bounded on both sides, any m-dimensional vector can be taken as Y.

The kernel Y generates auxiliary problem (C_Y). Without loss of generality, we may take all the components $b_i^{(Y)}$ of the constraint vector $B^{(Y)}$ of problem (C_Y) to be nonnegative. The vector with the components $\xi_j=0$, $j\in E_Y$, $\varepsilon_i=b_i^{(Y)}$, $i=1, 2, \ldots, m$ is, therefore, a support program of problem (C_Y) and its basis consists of unit vectors $e_1, e_2, \ldots, e_m$. Taking this program as the first, we solve the auxiliary problem (C_Y) by the simplex method (the second algorithm) for problems with bilateral restraints (see Chapter 4, § 5, and Chapter 5, 7-3).

After a finite number of steps, we obtain the support solution

$$\xi^*_j,\ j\in E_Y,\ \varepsilon^*_i,\quad i=1, 2, \ldots, m,$$

of problem (C_Y) and at the kernel $M^*=(\mu_1^*, \ldots, \mu_m^*)$ of the optimal program of problem $(\bar{C}_Y)$. Applying (4.17), we form quasiprogram X and proceed to analyze it.

If the residue ε_0^* of quasiprogram X is zero (case (a)), the quasiprogram X solves problem (A). Otherwise, the parameters δ_j^* are fomed (Δ_j have been previously computed while forming the set E_Y) and conditions (4.19) are checked. If these conditions are not satisfied (case (b)), problem (A) is proved unsolvable. In either of the two above cases the process of solution naturally terminates.

Now let $\varepsilon_0^* > 0$ and let Δ_j and δ_j^* have the same signs for some $j \in E_Y$, i.e., $\Delta_j \delta_j^* > 0$ (case (c)). In this case a new kernel of a program of problem $(\tilde{A})$ is formed and we proceed with the next iteration. As the new kernel we take the vector

$$Y' = Y - \theta_0 M^*,$$

where θ_0 is obtained from (4.25).

The kernel Y' defines a new auxiliary problem $(C_{Y'})$. The next iteration starts with the solution of this problem. As the basis of the first program of problem $(C_{Y'})$, we take the basis of the support solution of problem (C_Y).

The next iteration is the same as the above with one of the following three outcomes: the solution of problem (A) is obtained (case (a)), unsolvability of problem (A) is established, or auxiliary problem $(C_{Y''})$ is formed with which the next iteration starts (case (c)). In the nondegenerate case, each iteration lowers the residue of the quasiprogram.

Observe that all the auxiliary problems will be nondegenerate if the augmented problem (B) is nondegenerate. In this case any two quasiprograms of problem (A) obtained in the process of its solution have different residues.

A one-to-one correspondence exists between each quasiprogram of problem (A) and some support program of the augmented problem (B); the value of the linear form (4.8) on this program is equal to the residue of the quasiprogram. In the process of solution we thus move over different support programs of problem (B).

Hence, the problem (A) is investigated by the Hungarian method in a finite number of iterations. This conclusion also applies to the degenerate case, provided the auxiliary problems are solved using the rules guaranteeing that cycling will not occur (Chapter 4, § 6).

Thus, a linear-programming problem with bilateral restraints is solved by the Hungarian method in a finite number of iterations, each of which is made up of several simplex iterations.

§ 5. The primal-dual algorithm

5-1. As we have seen, to each iteration of the Hungarian method corresponds some feasible program Y of the dual problem $(\tilde{A})$. The program Y, in turn, defines the auxiliary problem (C_Y) of the current iteration. The solution of the auxiliary problem gives a quasiprogram X of the primal problem (A). The optimality criterion, which is in fact the basis of the method, enables us to test whether the quasiprogram X is a solution of problem (A). The unsolvability test gives a sufficient condition for problem (A) to have no programs. If quasiprogram X is not the optimal program

and there is no reason to suppose that the problem is unsolvable, elementary transformation defined by the solution of the two auxiliary problems (C_Y) and $(\tilde{C}_Y)$ is used to pass to a new program Y' of the dual problem $(\tilde{A})$. The next, successive iteration then starts with program Y' and proceeds as before.

In each step of the Hungarian method the auxiliary problem (C_Y) has to be solved. In general, problem (C_Y) can be solved either by the simplex method or by the dual simplex method. In each of the two methods the solution can be obtained either by the first or the second algorithm. Certain considerations, however, make it advisable to solve the auxiliary problem by the second algorithm of the simplex method. The simplex method is chosen because determination of the first program of each auxiliary problem does not involve special computations. The basis of the first program of the first auxiliary problem may be taken as the set of unit vectors $e_1, \ldots, e_m$. The basis components of the first support program of problem (C_Y) then coincide with the corresponding components b_i of the constraint vector B (b_i can always be taken as nonnegative). The first program of each successive auxiliary problem may be taken as the solution of the preceding auxiliary problem.

To proceed with the successive iteration in the Hungarian method, we require not only the solution of auxiliary problem (C_Y), but also the optimal program of its dual problem $(\tilde{C}_Y)$. The auxiliary problem is, therefore, best solved by the second simplex algorithm, which, besides the optimal program of the primal problem, yields also the solution of the dual problem. The lowest row of the last principal tableau of the second algorithm gives the components of the optimal program of the dual problem.

Application of the first algorithm is complicated by the fact that to form the successive auxiliary problem we must resolve all its restraint vectors not entering the preceding auxiliary problem in terms of basis vectors. This is unnecessary when using the second algorithm.

From the above it is obvious that the computational procedure of the Hungarian method should be based on the second simplex algorithm.

Solution of a problem by the Hungarian method thus involves successive filling in of several principal tableaus and one auxiliary tableau.

We will now describe the structure of the auxiliary tableau (Table 7.2).

In the upper part of the auxiliary tableau (the first $m+1$ rows) we write the components of the augmented restraint vectors of primal problem (A). When solving each auxiliary problem (C_Y), we use only those columns A_j in Table 7.2 for which $j \in E_Y$. It should be remembered that in auxiliary problems the coefficients of x_j in the linear form are all zero.

When solving auxiliary problems, the reader is advised to follow the following rule. Any unit vector e_s eliminated from the basis in one of the steps should not be included in successive bases. This restriction may, sometimes, yield a somewhat higher minimum of the linear form of an individual auxiliary problem. In other words, this restriction can lead to quasiprograms with exaggerated residue. Nevertheless, in general, this restriction reduces the laboriousness of computations, since it eliminates the necessity of examining the unit vectors $e_1, e_2, \ldots, e_m$ in the process of solution. The unit vectors $e_1, \ldots, e_m$ entering the system of restraint vectors of any auxiliary problem should, therefore, not be included in Table 7.2. We leave it to the reader to show that this restriction in the choice

TABLE 7.2

No.	B	A_1	A_2	$\cdots$	A_j	$\cdots$	A_n	Y
1	b_1	a_{11}	a_{12}	$\cdots$	a_{1j}	$\cdots$	a_{1n}	y_1
2	b_2	a_{21}	a_{22}	$\cdots$	a_{2j}	$\cdots$	a_{2n}	y_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
i	b_i	a_{i1}	a_{i2}	$\cdots$	a_{ij}	$\cdots$	a_{in}	y_i
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
m	b_m	a_{m1}	a_{m2}	$\cdots$	a_{mj}	$\cdots$	a_{mn}	y_m
$m+1$	c	c_1	c_2	$\cdots$	c_j	$\cdots$	c_n	
1	Δ	×	Δ_2	$\cdots$	Δ_j	$\cdots$	×	
	δ	δ_1	—	$\cdots$	—	$\cdots$	δ_n	
	δ'	δ_1'	—	$\cdots$	—	$\cdots$	δ_n'	
	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$	
	δ^*	δ_1^*	δ_2^*	$\cdots$	δ_j^*	$\cdots$	δ_n^*	
	θ	—	θ_2	$\cdots$	θ_j	$\cdots$	—	$\theta_0^{(1)}$
2	Δ	×	Δ_1	$\cdots$	×	$\cdots$	×	
	δ	δ_1	—	$\cdots$	δ_j	$\cdots$	δ_n	
	δ'	δ_1'	—	$\cdots$	δ_j'	$\cdots$	δ_n'	
	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$	
	δ^*	δ_1^*	δ_2^*	$\cdots$	δ_j^*	$\cdots$	δ_n^*	
	θ	—	θ_2	$\cdots$	—	$\cdots$	—	$\theta_0^{(2)}$
3	Δ	Δ_1	×	$\cdots$	×	$\cdots$	×	
	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$	

of vectors introduced into the basis does not affect the convergence in the Hungarian method (see Exercise 5).

In every iteration, the rows Δ, δ, δ', ..., δ^*, θ are filled in the lower part of the auxiliary tableau. The part of the auxiliary tableau comprising the rows Δ, δ, δ', ..., δ^*, θ will be called the auxiliary subtableau corresponding to the given iteration.

In the first iteration of the Hungarian method, the elements of row Δ are computed from the formula

$$\Delta_j = \sum_{i=1}^{m} a_{ij} y_i - c_j, \qquad j = 1, 2, \ldots, n, \tag{5.1}$$

where $Y = (y_1, \ldots, y_m)$ is the first program of the dual problem $(\tilde{A})$. In subsequent iterations the elements of row Δ are computed using recurrence formula (1.23). The positions j corresponding to zero elements of row Δ (marked by X) give the set E_Y of the indices of the vectors A_j entering the system of restraint vectors of the auxiliary problem (C_Y). The rows δ, δ', ..., δ^* give evaluations of the restraint vectors A_j $(j \in E_Y)$ of the auxiliary problem (C_Y) relative to the corresponding basis,

$$\delta_j^{(l)} = \sum_{i=1}^{m} a_{ij} \mu_i^{(l)} \quad \text{for} \quad j \in E_Y \tag{5.2}$$

(the coefficients of x_j in the linear form of the auxiliary problem are all zero). Here the parameters $\mu_i^{(l)}$ are evaluations of the restraints of the auxiliary problem relative to its program $(\Xi^{(l)}, E^{(l)})$. In the process of solution of the auxiliary problem the elements $\delta_j^{(l)}$ of rows $\delta^{(l)}$ are computed for $j \in E_Y$ only. The remaining positions of $\delta^{(l)}$ are crossed out. The only exception is the row δ^* corresponding to the optimal program of the auxiliary problem. The elements δ_j^* are computed from formulas (1.22) for all j $(j = 1, 2, \ldots, n)$.

In the last row of the auxiliary subtableau corresponding to each iteration only those positions are filled which correspond to j for which $\delta_j^* > 0$. The elements of the row θ are computed as ratios of the corresponding elements of the rows Δ and δ^*. The least element θ_0 of row θ is taken as the parameter of the elementary transformation of program Y of problem $(\tilde{A})$ into program Y'.

In the auxiliary tableau given in this article it is assumed, to be specific, that in the first iteration only Δ_1 and Δ_n vanish in row Δ, whereas in the second iteration $\Delta_j = 0$ also. In row θ cells corresponding to nonpositive δ_j^* are crossed out.

The elements of the subtableaus are denoted by three indices: the number of the iteration using the Hungarian method, the number of the iteration in the solution of the auxiliary problem, and the number of the vector. To avoid complicated notations, we omit indices whenever no confusion, can arise.

As we see, the structure of the auxiliary tableau using the Hungarian method is somewhat different from the structure of the corresponding tableau using the second simplex algorithm. In each iteration of the simplex method one row is added to the auxiliary tableau, whereas in each iteration of the Hungarian method a subtableau, consisting usually of several rows, is added to the auxiliary tableau.

The principal tableaus in the Hungarian method (see Table 7.3) are constructed in the same way as in the second simplex algorithm.

TABLE 7.3

	No.	σ	B	e_0	e_1	e_2	...	e_l	...	e_m	A_k	θ	No. of tableau
	1	1	e_1	b_1	1		...		...		a_{1k}	θ_1	
	2	1	e_2	b_2		1	...		...		a_{2k}	θ_2	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	...	$\vdots$	$\vdots$	$\vdots$	
$\longleftarrow$	r	1	e_r	b_r			...		...		a_{rk}	θ_r	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	...	$\vdots$	$\vdots$	$\vdots$	
	m	1	e_m	b_m			...		...	1	a_{mk}	θ_m	
	$m+1$	—	—	$e_{m+1,0}$	1	1	...	1	...	1	δ_k^*	—	
	1	1	e_1	e'_{10}	e'_{11}	e'_{12}	...	e'_{1l}	...	e'_{1m}	$z'_{1k'}$	θ'_1	
	2	1	e_2	e'_{20}	e'_{21}	e'_{22}	...	e'_{2l}	...	e'_{2m}	$z'_{2k'}$	θ'_2	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	...	$\vdots$	$\vdots$	$\vdots$	
	$r-1$	1	e_{k-1}	$e'_{r-1,0}$	$e'_{r-1,1}$	$e'_{r-1,2}$	...	$e'_{r-1,l}$	...	$e'_{r-1,m}$	$z'_{r-1,k'}$	θ'_{k-1}	
$\longrightarrow$	r		A_k	e'_{r0}	e'_{r1}	e'_{r2}	...	e'_{rl}	...	e'_{rm}	$z'_{rk'}$	θ'_r	1′
	$r+1$	1	e_{r+1}	$e'_{r+1,0}$	$e'_{r+1,1}$	$e'_{r+1,2}$	...	$e'_{r+1,l}$	...	$e'_{r+1,m}$	$z'_{r+1,k'}$	θ'_{r+1}	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	...	...	...	$\vdots$	...	$\vdots$	$\vdots$	$\vdots$	
	m	1	e_m	e'_{m0}	e'_{m1}	e'_{m2}	...	e'_{ml}	...	e'_{mm}	$z'_{mk'}$	θ'_m	
	$m+1$	—	—	$e'_{m+1,0}$	$e'_{m+1,1}$	$e'_{m+1,2}$	...	$e'_{m+1,l}$	...	$e'_{m+1,m}$	$\delta^*_{k'}$	—	

TABLE 7.3 (continued)

	No.	σ	B	e_0	e_1	e_2	...	e_i	...	e_m	A_k	θ	No. of tableau
	1	1	e_1	e''_{10}	e''_{11}	e''_{12}	...	e''_{1i}	...	e''_{1m}	$z''_{1k''}$	θ''_1	$1''$
	2	1	e_2	e''_{20}	e''_{21}	e''_{22}	...	e''_{2i}	...	e''_{2m}	$z''_{2k''}$	θ''_2	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	...	$\vdots$	$\vdots$	$\vdots$	
	m	1	e_m	e''_{m0}	e''_{m1}	e''_{m2}	...	e''_{mi}	...	e''_{mm}	$z''_{mk''}$	θ''_m	
	$m+1$	—	—	$e''_{m+1,0}$	$e''_{m+1,1}$	$e''_{m+1,2}$	...	$e''_{m+1,i}$	...	$e''_{m+1,m}$	$\delta^*_{k''}$	—	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	...	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	1	1	e_1	e_{10}	e_{11}	e_{12}	...	e_{1i}	...	e_{1m}	$z_{1\bar{k}}$	θ_1	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	...	$\vdots$	$\vdots$	$\vdots$	
	m	1	e_m	e_{m0}	e_{m1}	e_{m2}	...	e_{mi}	...	e_{mm}	$z_{m\bar{k}}$	θ_m	
	$m+1$	—	—	$e_{m+1,0}$	$e_{m+1,1}$	$e_{m+1,2}$	...	$e_{m+1,i}$	...	$e_{m+1,m}$	$\delta^*_{\bar{k}}$	—	
	1	1	e_1	e'_{10}	e'_{11}	e'_{12}	...	e'_{1i}	...	e'_{1m}	$z'_{1\bar{k}'}$		$2'$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	...	$\vdots$	$\vdots$	$\vdots$	
	m	1	e_m	e'_{m0}	e'_{m1}	e'_{m2}	...	e'_{mi}	...	e'_{mm}	$z'_{m\bar{k}'}$		
	$m+1$	—	—	$e'_{m+1,0}$	$e'_{m+1,1}$	$e'_{m+1,2}$	...	$e'_{m+1,i}$	...	$e'_{m+1,m}$	$\delta^*_{\bar{k}'}$	—	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$	...	$\vdots$	$\vdots$	$\vdots$	$\vdots$

The row Δ of the auxiliary tableau corresponding to the first program Y of the dual problem $(\tilde{A})$ defines the set of indices E_Y and enables us to form the first auxiliary problem (C_Y).

Problem (C_Y) is solved following the usual rules of the second simplex algorithm. Principal tableaus are filled successively, and after each step in the process of solution of problem (C_Y) we add a row $\delta^{(l)}$ to the auxiliary subtableau l. Entries are, obviously, written only in those positions $\delta_j^{(l)}$ for which $j \in E_Y$.

The main part of the first principal tableau is filled without any special computations. The unit vectors $e_1, e_2, \ldots, e_m$ appearing in the system of restraint vectors of the first auxiliary problem should be taken as its first basis. Therefore

$$e_{ij} = \begin{cases} 0 \text{ for } i \neq j, \\ 1 \text{ for } i = j, \end{cases} \quad i, j = 1, 2, \ldots, m,$$

$$e_{i0} = b_i, \quad i = 1, 2, \ldots, m,$$

$$e_{m+1,0} = \sum_{i=1}^{m} e_{i0} = \sum_{i=1}^{m} b_i,$$

$$e_{m+1,j} = \mu_j = \sum_{i=1}^{m} \sigma_i e_{ij} = 1, \quad j = 1, 2, \ldots, m.$$

Here, $\sigma_1 = \sigma_2 = \ldots = \sigma_m = 1$ are the coefficients of $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$ in the linear form of the auxiliary problem.

Columns A_k and θ in the principal zeroeth tableau are filled according to the general rules of the second simplex algorithm; this algorithm is also used to pass to successive principal tableaus. (The reader should not confuse the notations θ and θ_0 of the last column in the principal tableau and of its least element, respectively, with analogous notations used to construct the auxiliary tableau.) In column σ of the principal tableau we write the coefficients of the basis variables of the linear form for the corresponding iteration of the auxiliary problem. The coefficients σ_i corresponding to the vectors e_i are all equal to unity. The coefficients corresponding to the vectors A_j $(j \in E_Y)$ appearing in the basis of the support program of problem (C_Y) are all zero.

The vector A_k to be introduced into the basis is determined by the maximum positive elements of row δ of the auxiliary tableau. The maximum positive δ (and not the least negative) is chosen because the auxiliary problem is to minimize the linear form and not to maximize it.

The reader is now advised to reread Chapter 5, § 5 where the process of solution by the second simplex algorithm, using the terminology and the notations of the auxiliary problem, is described.

After several simplex iterations we have the row $\delta^{(\nu)}$ in the auxiliary tableau which contains nonpositive entries only (this refers only to those positions in the row which correspond to the restraint vectors of the auxiliary problem). In Table 7.2 this row is marked as δ^*.

The elements e_{i0}^* in column e_0 of the last principal tableau, corresponding to the basis vectors A_{s_i} $(s_i \in E_Y)$, define a quasiprogram X of problem (A). If $e_{m+1,0}^*$ (the residue of quasiprogram X) in the lower left corner of the tableau is zero, quasiprogram X is the optimal program of problem (A). The process of solution then terminates (case (a)).

Now let $e_{m+1,0}^* \neq 0$. In this case, using formula (1.22) we fill in the empty cells in row δ^*. If all the new entries δ_j^* are nonpositive then, according

to the unsolvability test (case (b)) the process of solution is terminated by establishing inconsistency of restraints of problem (A). If some of the δ_j^* are positive (case (c)), we proceed with the next iteration. To this end we fill in the cells of row θ corresponding to $\delta_j^* > 0$ appearing in the ratios Δ_j/δ_j^* and select the least element θ_*. Further, using the relationship

$$Y' = Y - \theta_* M^*$$

we compute the successive program of problem $\tilde{A}$, which is used as the point of departure in the next iteration. However, there is no need to

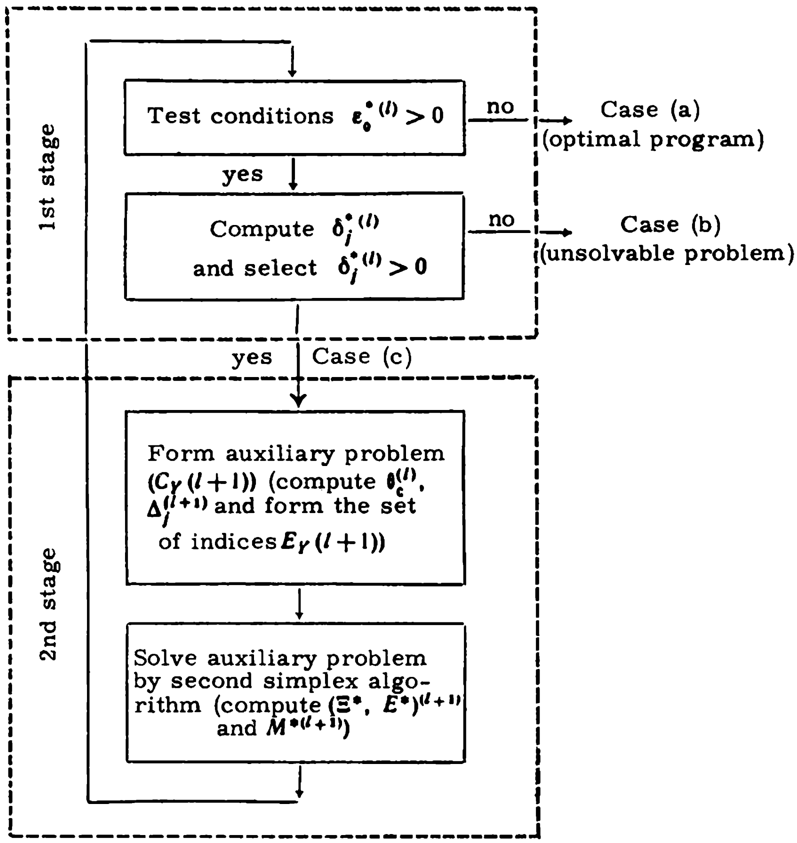


FIGURE 7.5

compute a program of problem $(\tilde{A})$ at every step. It is better to compute the first row of the subtableau pertaining to the successive iteration in the Hungarian method using the formula

$$\Delta' = \Delta(\theta_*) = \Delta - \theta_* \delta^*. \tag{5.3}$$

The zero entries of the row Δ' define a new set $E_{Y'}$ and thus a successive auxiliary problem $(C_{Y'})$.

The process is then resumed following the same rules as in the first iteration.

It should be emphasized that the main part of the first principal tableau in each successive iteration is taken as the main part of the last principal tableau in the preceding iteration.

The iterations are continued until either case (a) or case (b) is obtained. In case (a) the problem is solved, and in case (b) the problem is shown to be unsolvable.

One more remark. The solution of each auxiliary problem does not require, as a rule, numerous iterations. This is so because the solution of the preceding auxiliary problem (C_Y), taken as the first support program of the next problem $(C_{Y'})$ is generally fairly close to the optimal program of problem $(C_{Y'})$. In solving the first auxiliary problem we start with a unit basis. Computation of the optimal program of the first auxiliary problem is not particularly laborious if the number of elements in the set E_Y is far less than m. If the set E_Y contains more elements (of the order of m) the solution of the auxiliary problem involves rather cumbersome computations. In this case, however, far less iterations are required to solve the primal problem by the Hungarian method.

Figure 7.5 shows a block diagram of the solution of a linear-programming problem by the Hungarian method.

5-2. We now illustrate the application of the algorithm by the following example.

Maximize the linear form

$$L(X) = x_1 + x_2 + x_3 + 3x_4 + x_5 + 3x_6 + x_7$$

subject to the conditions

$$\begin{aligned} x_1 + 2x_2 + 2x_3 + 3x_4 + 2x_5 + 4x_6 + 2x_7 &= 5, \\ 2x_1 + x_2 + 4x_3 - x_4 + x_5 + 2x_6 - 3x_7 &= 7, \\ x_1 + 3x_2 + x_3 + 4x_4 - x_5 + 3x_6 + 2x_7 &= 5, \\ -x_1 + 2x_2 + x_3 + x_4 + 2x_5 - x_6 + x_7 &= 2, \\ x_j \geq 0, \quad j = 1, 2, \ldots, 7. \end{aligned}$$

All the coefficients of the first restraint are positive. According to Chapter 6, 7-2, the components of the first program of the dual problem can be determined as follows:

$$y_1 = \max\left\{\frac{1}{1}, \frac{1}{2}, \frac{1}{2}, \frac{3}{3}, \frac{1}{2}, \frac{3}{4}, \frac{1}{2}\right\} = 1, \quad y_2 = y_3 = y_4 = 0.$$

We form an auxiliary tableau. In row Δ we write

$$\Delta_j = \sum_{i=1}^{4} a_{ij} y_i - c_j, \qquad j = 1, 2, \ldots, 7.$$

Two elements in row Δ are zero (Δ_1 and Δ_4). Hence, auxiliary problem (C_Y) contains, besides the unit restraint vectors e_i $(i = 1, \ldots, 4)$, two vectors only, A_1 and A_4. We fill in the main part of the first principal tableau. In the last row, in positions $e_1, \ldots, e_4$ we write the relative evaluations of restraints of problem (C_Y). We have

$$\mu_i = 1, \qquad i = 1, 2, 3, 4.$$

We then compute the relative evaluations of the restraint vectors of problem (C_Y). We have

$$\delta_1 = \sum_{i=1}^{4} a_{i1}\mu_i = 3; \quad \delta_4 = 7.$$

The evaluations of vectors A_1 and A_4 are positive. A_4 with the higher evaluation is introduced into the basis of problem (C_Y). The vector A_4 is resolved in the vectors of the first basis and the coefficients (coinciding with the components of A_4) are written in column A_k of the principal tableau. The $(m+1)$-th (fifth) entry in column A_k is $\delta_4 = 7$. Then, for nonnegative entries of column A_k we compute the components of column θ, i. e., the ratios of the corresponding elements in columns e_0 and A_k. The last entry $\theta_0 = {}^5/_4$ in column θ is obtained on the basis vector e_3. The vector e_3 is to be eliminated from the basis.

TABLE 7.4

No.	B	A_1	A_2	A_3	A_4	A_5	A_6	A_7	Y
1	5	1	2	2	3	2	4	2	1
2	7	2	1	4	−1	1	2	−3	
3	5	1	3	1	4	−1	3	2	
4	2	−1	2	1	1	2	−1	1	
5	c	1	1	1	3	1	3	1	
1	Δ	×	1	1	×	1	1	1	
	δ	3	—	—	**7**	—	—		
	δ'	**5/4**	—	—		—	—		
	δ^*		$^{16}/_{9}$	$^{35}/_{9}$		$^{16}/_{3}$	$^{11}/_{9}$	$-^{1}/_{9}$	
	θ	—	$^{9}/_{16}$	$^{9}/_{35}$	—	**3/16**	$^{9}/_{11}$	—	$\theta_0^{(1)}={}^{3}/_{16}$
2	Δ	×	$^{2}/_{3}$	$^{13}/_{48}$	×	×	$^{37}/_{48}$	$^{49}/_{48}$	
	δ^*		$^{8}/_{3}$	$^{7}/_{3}$			$-^{5}/_{3}$	$-^{5}/_{3}$	
	θ	—	$^{1}/_{4}$	**13/112**	—	—	—	—	$\theta_0^{(2)}={}^{13}/_{112}$
3	Δ	×	$^{5}/_{14}$	×	×	×	$^{27}/_{28}$	$^{17}/_{14}$	
	δ^*		4			−8	−6	−4	
	θ	—	**5/56**	—	—	—	—	—	$\theta_0^{(3)}={}^{5}/_{56}$
4	Δ	×	×	×	×	$^{5}/_{7}$	$^{3}/_{2}$	$^{11}/_{7}$	

TABLE 7.5 (1–4)

	No.	σ	B	e_0	e_1	e_2	e_3	e_4	A_k	θ	No. of tableau
	1	1	e_1	5	1				3	$5/3$	
	2	1	e_2	7		1			-1	—	
←	3	1	e_3	5			1		4	**5/4**	1
	4	1	e_4	2				1	1	2	
	5	μ_0	—	19	1	1	1	1	7	—	
	1	1	e_1	$5/4$	1		$-3/4$		$1/4$	5	
←	2	1	e_2	$33/4$		1	$1/4$		$9/4$	**11/3**	
→	3		A_4	$5/4$			$1/4$		$1/4$	5	1′
	4	1	e_4	$3/4$			$-1/4$	1	$-5/4$	—	
	5	μ_0'	—	$41/4$	1	1	$-3/4$	1	$5/4$	—	
←	1	1	e_1	$1/3$	1	$-1/9$	$-7/9$		$5/3$	**1/8**	
→	2		A_1	$11/3$		$4/9$	$1/9$		$1/3$	11	
	3		A_4	$1/3$		$-1/9$	$2/9$		$-1/3$	—	1″
	4	1	e_4	$16/3$		$5/9$	$-1/9$	1	$8/3$	2	
	5	μ_0''	—	$17/3$	1	$4/9$	$-8/9$	1	$16/3$	—	

TABLE 7.5 (continued)

	No.	σ	B	e_0	e_1	e_2	e_3	e_4	A_k	θ	No. of tab-leau
⇄	1		A_5	1/8	3/8	−1/24	−7/24		7/24	**3/7**	2
	2		A_1	29/8	−1/8	11/24	5/24		43/24	87/43	
	3		A_4	3/8	1/8	−1/8	1/8		1/8	3	
	4	1	e_4	5	−1	2/3	2/3	1	7/3	15/7	
	5	μ_1	—	5	−1	2/3	2/3	1	7/3	—	
→	1		A_3	3/7	9/7	−1/7	−1		−4/7	—	3
	2		A_1	20/7	−17/7	5/7	2		13/7	20/13	
	3		A_4	8/7	2/7	−1/7			8/7	1	
←	4	1	e_4	4	−4	1	3	1	4	**1**	
	5	μ_2	—	4	−4	1	3	1	4	—	
	1		A_3	1	5/7		−4/7	1/7			4
	2		A_1	1	−4/7	1/4	17/28	−13/28			
	3		A_4		5/7	−1/4	−9/28	−3/28			
→	4		A_2	1	−1	1/4	3/4	1/4			
	5	μ_3	—							—	

Resuming the process of solution of the auxiliary problem according to the rules of the second simplex algorithm, we obtain after two iterations the solutions of problem (C_Y) and $(\tilde{C}_Y)$:

$$(\Xi^*, E^*) = \left(\frac{11}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, \frac{16}{3}\right),$$

$$M^* = \left(1, \frac{4}{9}, -\frac{8}{9}, 1\right).$$

The residue of the corresponding quasiprogram is 17/3. Hence, this quasiprogram is not the solution of the problem.

Filling in the unoccupied cells in row δ^* of the auxiliary tableau, we obtain

$$\delta^* = \left(0, \frac{16}{9}, \frac{35}{9}, 0, \frac{16}{3}, \frac{11}{9}, -\frac{1}{9}\right).$$

In the last row θ of the subtableau corresponding to the first iteration we write four elements only (in cells where $\delta_j^* > 0$). The least element θ_0 in row θ is 3/16.

We now proceed with the second iteration. The elements of the first row of the subtableau corresponding to the second iteration are computed from the formula

$$\Delta_j' = \Delta_j - \theta_0 \delta_j^*.$$

Row Δ of subtableau 2 contains three null entries, Δ_1', Δ_4', Δ_5'. Hence, besides the unit vectors e_i $(i = 1, 2, 3, 4)$, the new auxiliary problem $(C_{Y'})$ contains the restraint vectors A_1, A_4, and A_5.

As the main part of the first principal tableau in the solution of problem $(C_{Y'})$ we take the last (third) principal tableau in the solution of problem (C_Y). The second iteration of the Hungarian method contains one step only. The solutions of the third and the fourth auxiliary problems $(C_{Y''})$ and $(C_{Y'''})$ are also obtained in one step each. The quasiprogram

$$X^* = (1, 1, 1, 0, 0, 0, 0)$$

corresponding to problem $(C_{Y'''})$ has a zero residue. Quasiprogram X^* is thus the optimal program of the problem.

The entire process of solution is represented in Table 7.5 (principal tableaus) and in Table 7.4 (auxiliary tableau).

5-3. Application of the simplex method presupposes knowledge of some support program of primal problem (A). When solving a linear-programming problem by the dual simplex method we start with some support program of the dual problem $(\tilde{A})$. Computation of a support program of the primal or the dual problem is in general no less tedious than determination of the optimal program from a given support program.

The Hungarian method, on the other hand, starts with any feasible (not necessarily support) program of the dual problem $(\tilde{A})$. This is a considerable asset, since in many cases the choice of a feasible nonsupport program of the dual problem does not require any special computations. If, however, the structure of the problem does not indicate a first feasible program, we must apply the method discussed in Chapter 6, 7-6. According to this method, problem (A) is supplemented by

$$\sum_{j=1}^{n} x_j \leqslant M,$$

or

$$\sum_{j=0}^{n} x_j = M, \quad x_0 \geqslant 0.$$

Here M is assumed sufficiently large. The resulting linear-programming problem is called problem (M). Its dual, problem $(\tilde{M})$ is to find the minimum of the linear form

$$\sum_{i=1}^{m} b_i y_i + M y_0,$$

whose variables are subject to the conditions

$$\sum_{i=0}^{m} a_{ij} y_i \geqslant c_j, \quad j = 1, 2, \ldots, n,$$

$$y_0 \geqslant 0.$$

As a feasible program of problem $(\tilde{M})$ we can, obviously, take the vector

$$\tilde{Y} = (y_0, 0, \ldots, 0),$$

where

$$y_0 = \max_{1 < j < n} \{c_j, 0\}.$$

We now solve problem (M) by the Hungarian method starting with program $\tilde{Y}$. In the process of solution the number M is assumed greater than any other number with which it is compared. Let

$$\tilde{X}^* = (x_0^*, x_1^*, \ldots, x_n^*)$$

be the solution of problem (M). Two cases are possible·

1. The index $j=0$ belongs to the set $E_Y(s)$, where $Y^{(s)}$ is the program of the dual problem corresponding to the last iteration of the Hungarian method, i. e., $\Delta_0^{(s)} = 0$.

2. The index $j=0$ does not belong to the set $E_Y(s)$, i. e., $\Delta_0^{(s)} > 0$.

In the first case, the vector

$$X^* = (x_1^*, x_2^*, \ldots, x_n^*)$$

is the optimal program of problem (A). In the second case problem (A) is unsolvable. Following the considerations given in Chapter 6, 7-6, the reader can prove this proposition (Exercise 9). Further, it can easily be shown that if the process of solution of problem (M) terminates in case (b) (problem (M) is unsolvable), the restraints of the primal problem are inconsistent (Exercise 10).

5-4. Let us now consider the special features of the Hungarian (or the primal-dual) algorithm in application to linear-programming problems with bilateral restraints.

The bilateral restraints force us to introduce certain modifications in the structure of the principal and the auxiliary tableaus. As with a problem in canonical form, each iteration of the Hungarian method involves filling several principal tableaus and a subtableau of the auxiliary tableau.

In the principal tableaus and the subtableau of the auxiliary tableau corresponding to an individual iteration of the Hungarian method, we give the process of solution of the successive auxiliary problem according to the second simplex algorithm. Each subtableau, .moreover, contains the rows Δ, (α, β), δ^*, and θ. In rows Δ, (α, β), $\delta^* = \delta^{(s)}$ and $(\alpha, \beta)^{(s)}$ (s is the number of the last simplex iteration in the solution of the auxiliary problem) all the elements are written (the cells in row Δ corresponding to $\Delta_j = 0$ are marked by crosses). In row θ only those elements are written for which the corresponding elements in rows Δ and δ^* have like signs.

The elements of row Δ in the first subtableau are computed from

$$\Delta_j = \sum_{i=1}^{m} a_{ij} y_i - c_j,$$

where y_i are the kernel components of a program of the dual problem $(\tilde{A})$. The elements of row Δ_j' of the next subtableau are computed from the

recurrence formula

$$\Delta_j' = \Delta_j - \theta_0 \delta_j^*.$$

Row Δ makes it possible to define the set E_Y and thus to determine the successive auxiliary problem. In row (α, β) following row Δ we write the symbol α when $\Delta_j > 0$ and the symbol β when $\Delta_j < 0$. The elements in row (α, β) for $j \in E_Y$ are computed following the rules for the solution of linear-programming problems with bilateral restraints according to the second simplex algorithm (see Chapter 5, 7-3).

In subsequent rows of the subtableau $(\delta, (\alpha, \beta)', \delta', (\alpha, \beta)'', \ldots, (\alpha, \beta)^{(s-1)}, \delta^{(s-1)})$ only entries which correspond to the restraint vectors of the problem, are made. In row $(\alpha, \beta)^{(s)}$ the elements for $j \in E_Y$ are written following the same rules as in other iterations with problem (C_Y). The elements for $j \notin E_Y$ give the same limits of problem variables as those indicated in row (α, β). The elements in row $(\alpha, \beta)^{(s)}$ contain the values of extrabasis components of the corresponding quasiprogram.

The elements in row δ^* are calculated according to

$$\delta_j^* = \sum_{i=1}^{m} a_{ij} \mu_i^*.$$

For $j \in E_Y$ $\delta_j^* = \delta_j^{(s)}$. The last row θ of the subtableau is used to determine the parameter θ_0 of the elementary transformation of kernel Y into kernel Y' of the successive program of the dual problem. Kernel Y' specifies the transition to the next iteration of the Hungarian method.

Successive subtableaus of the auxiliary tableau are filled following the same rules. The only difference is, as we have already observed, that the row $\Delta^{(v)}$ is no longer computed directly, but from a recurrence formula in terms of the parameters of the preceding iteration.

To complete the description of the auxiliary tableau we note that in the upper part column $B^{(Y)}$ is added to the left, column Y to the right, and row (α/β) above. Column $B^{(Y)}$ contains the components

$$b_i^{(Y)} = b_i - \sum_{j \notin E_Y} \gamma_j A_j$$

of the constraint vector of the first auxiliary problem (γ_j are computed from (4.6)). The elements of row Y are the kernel components of the first program of the dual problem. The elements in the upper row (α/β) give the upper and the lower limits of variation of the problem variables.

All the special features of principal tableaus connected with the solution of auxiliary problems (each of these is a problem with bilateral restraints) are discussed in Chapter 5, 7-3. Here we note a certain inconsistency in the notations of Chapters 5 and 7, which need not cause any confusion. In Chapter 5 the relative evaluations of problem restraints are denoted by λ_i. In Chapter 7 the preliminary evaluations of the restraints of the auxiliary problem are denoted by μ_i, and the components of the solution of problem $(\bar{C}_Y)$, the dual of problem (C_Y), are denoted by μ_i^*.

The principal tableaus contain column θ whose elements are computed from formula (7.2) of Chapter 5. The elements of row θ in subtableaus of the auxiliary tableau are defined by (4.24) in this chapter.

When filling the principal tableaus, the following should be observed.

TABLE 7.6

$B^{(Y)}$	No.	B	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	Y
		α/β	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	
5	1	7	1	1	2	3	2	3	1	1				0.8
5	2	8	2	3	1	1	3	2	2		1			
5.5	3	6	1	2	3	2	0,5	1	1			1		0.6
5	4	7	2	1	3	1	2	3	1				1	
	5	c	1	2	3	1	2	3	1	−1	−1	−1	−1	
	1	Δ	0.4	×	0.4	2.6	−0.1	×	0.4	1.8	1	1.6	1	
		(α, β)	α	α	α	α	β	α	α	α	α	α	α	
		δ	—	7	—	—	—	**9**	—	—	—	—	—	
		(α, β)′	—	α	—	—	—	β	—	—	—	—	—	
		δ′	—	**7**	—	—	—	−9	—	—	—	—	—	
		(α, β)″	α	—	α	α	β	β	α	α	α	α	α	
		δ*	1.333		6.667	4.667	0.5	4.333	0.333	1	−1.333	1	1	
		θ	0.3	—	**0.06**	0.557	—	—	1.2	1.8	—	1.6	1	$\theta_0^{(1)}=0.06$
	2	Δ	0.32	×	×	2.32	−0.13	−0.26	0.38	1.74	1.08	1.54	0.94	
		(α, β)	α	—	—	α	β	β	α	α	α	α	α	
		δ*	−2			3	−2	−1.5	−0.5	1	−0.5	1	−1.5	
		θ	—	—	=	0.774	**0.065**	0.173	—	1.74	—	1.54	—	$\theta_0^{(2)}=0.065$

TABLE 7.6 (continued)

$B^{(Y)}$ No.	α/β / B	$^0/_1$ A_1	$^0/_1$ A_2	$^0/_1$ A_3	$^0/_1$ A_4	$^0/_1$ A_5	$^0/_1$ A_6	$^0/_1$ A_7	$^0/_1$ A_8	$^0/_1$ A_9	$^0/_1$ A_{10}	$^0/_1$ A_{11}	Y
3	Δ	0.45	×	×	2.125	×	−0.162	0.413	1.675	1.112	1.475	1.038	
	(α,β)	α	β	—	α	—	β	α	α	α	α	α	
	δ^*	−0.857	2.286		3.571		−0.643	0.929	1	0.357	1	−1.786	
	θ	—	—	—	0.595	—	**0.253**	0.444	1.675	3.115	1.475	—	$\theta_0^{(3)}=0.253$
4	Δ	0.667	−0.578	×	1.222	×	×	0.178	1.422	1.022	1.222	1.489	
	(α,β)	α	β	—	α	β	—	α	α	α	α	α	
	δ^*		4		4	1.5		2	1	1	1	−2	
	θ	—	—	—	0.306	—	—	**0.0889**	1.422	1.022	1.222	—	$\theta_0^{(4)}=0.0889$
5	Δ	0.667	−0.933	×	0.867	−0.133	×	×	1.333	0.933	1.133	1.667	
	(α,β)	α	β	—	α	β	—	—	α	α	α	α	
	δ^*	−1	0.2		2.6	−0.4			1	−0.2	0.4	−1	
	θ	—	—	—	**0.333**	0.333	—	—	1.333	—	2.833	—	$\theta_0^{(5)}=0.333$
6	Δ	1	−1	×	×	×	×	×	1	1	1	2	
	(α,β)	α	β	—	—	β	—	α	α	α	α	α	
	δ	—	—			**−3.929**		−3.714	—	—	—	—	
	$(\alpha,\beta)'$	α	β	—	—	—	—	α	α	α	α	α	
	δ'	0	0	0	0	0	0	0	0	0	0	0	

TABLE 7.7 (1–6')

	No.	δ	B_X	e_0	e_1	e_2	e_3	e_4	A_k	$\tilde{A}_k$	γ	θ	No. of tableau
	1	1	e_1	5	1				3	3	0	1.667	1
	2	1	e_2	5		1			2	2	0	2.5	
	3	1	e_3	5.5			1		1	1	0	5.5	
	4	1	e_4	5				1	3	3	0	1.667	
←	5	$A_k = A_6$		0	—	—	—	—	—1	—1	1	1	
	6	—	μ_1	20.5	1	1	1	1	9	9	—	—	
	1	1	e_1	2	1				1	1	0	2	1′
←	2	1	e_2	3		1			3	3	0	1	
	3	1	e_8	4,5			1		2	2	0	2.25	
	4	1	e_4	2				1	1	1	0	2	
	5	$A_k = A_2$		0	—	—	—	—	—1	—1	1	1	
	6	—	μ_1'	11.5	1	1	1	1	7	7	—	—	
	1	1	e_1	1	1	—0.333			1.667	1.667	0		1″
→	2		A_2	1		0.333			0.333	0.333	0		
	3	1	e_3	2.5		—0.667	1		2.333	2.333	0		
←	4	1	e_4	1		—0.333		1	2.667	2.667	0		
	5	$A_k = A_3$		0	—	—	—	—	—1	—1	1		
	6	—	μ_1''	4.5	1	—1.333	1	1	6.667	6.667	—		

TABLE 7.7 (continued)

	No	δ	B_X	e_0	e_1	e_2	e_3	e_4	A_k	$\tilde{A}_k$	γ	θ	No. of tableau
	1	1	e_1	0.375	1	−0.125		−0.625	0.375	−0.375	—	—	2
←	2		A_2	0.875		0.375		−0.125	0.875	−0.875	1	**0.143**	
	3	1	e_3	1.625		−0.375	1	−0.875	−2.375	2.375	0	0.168	
→	4		A_3	0.375		−0.125		0.375	0.375	−0.375	1	1.667	
	5	$A_k = A_5$		1	—	—	—	—	−1	1	0	1	
	6	—	μ_2	2	1	−0.5	1	−1.5	−2	2	—	—	
	1	1	e_1	0.429	1	−0.286		−0.571	0.714	−0.714	—	—	3
→ ←	2		A_5	0.857		0.429		−0.143	0.429	−0.429	1	**0.333**	
	3	1	e_3	1.286		0.643	1	−1.214	−1.357	1.357	0	0.947	
	4		A_3	0.429		−0.286		0.429	0.714	−0.714	1	0.8	
	5	$A_k = A_6$		1	—	—	—	—	−1	1	0	1	
	6	—	μ_3	1.714	1	0.357	1	−1.786	−0.643	0.643	—	—	
	1	1	e_1	0.667	1	−1		−0.333	−1.333	−1.333	—	—	4
→	2		A_6	0.667		1		−0.333	1.667	1.667	0	0.4	
←	3	1	e_3	0.833		2	1	−1.667	3.333	3.333	0	0.25	
	4		A_2	0.667		−1		0.667	−1.333	−1.333	1	0.25	
	5	$A_k = A_7$		0	—	—	—	—	−1	−1	1	1	
	6	—	μ_4	1.5	1	1	1	−2	2	2	—	—	

TABLE 7.7 (continued)

	No.	σ	B_χ	e_0	e_1	e_2	e_3	e_4	A_k	$\tilde{A}_k$	γ	θ	No. of tableau
	1	1	e_1	1	1	−0.2	0.4	−1	2.6	2.6	0	0.385	5
	2		A_6	0.25			−0.5	0.5	−0.5	−0.5	1	1.5	
→ ←	3		A_7	0.25		0.6	0.3	−0.5	0.7	0.7	0	0.357	
	4		A_3	1		−0.2	0.4		0.6	0.6	0	1,667	
	5	$A_k=A_4$		0	—	—	—	—	−1	−1	1	1	
	6	—	μ_5	1	1	−0.2	0.4	−1	2.6	2.6	←	—	
←	1	1	e_1	0.0714	1	−2.429	−0.714	0.857	−3.929	3.929	0	0.0182	6
	2		A_6	0.429		0.429	−0.286	0.143	1.429	−1.429	1	0.4	
→	3		A_4	0.357		0.857	0.429	−0.714	1.357	−1.357	1	0.474	
	4		A_3	0.786		−0.714	0.143	0.429	−1.214	1.214	0	0.647	
	5	$A_k=A_5$		1	—	—	—	—	−1	1	0	1	
	6	—	μ_6	0.0714	1	−2.429	−0.714	0.857	−3.929	3.929	—	—	
→	1		A_5	0.982	−0.255	0.618	0.182	−0.218					6′
	2		A_6	0.455	0.364	−0.455	−0.545	0.455					
	3		A_4	0.382	0.345	0.0182	0.182	−0.418					
	4		A_3	0.764	−0.309	0.0364	0.364	0.164					
	5				—	—	—	—					
	6	—	μ'_6	0	0	0	0	0			—	—	

Column e_0 of the principal tableaus contains the coefficients of the expansion of the vector

$$e_0 = B^{(Y)} - \sum_{\substack{j \in E_Y \\ j \notin I_X}} \gamma_j A_j = B - \sum_{j \in E_Y} \gamma_j A_j - \sum_{\substack{j \in E_Y \\ j \in I_X}} \gamma_j A_j = B - \sum_{j \in I_X} \gamma_j A_j \qquad (5.4)$$

in terms of basis vectors of the support program of problem (C_Y). Here the parameters γ_j are computed from (4.6), I_X is the set of indices of vectors A_j entering the basis of the support program of the corresponding auxiliary problem.

When solving the auxiliary problem, the elements of the main parts of the principal tableaus are transformed according to formula (7.7) of Chapter 5. The solution of problem (C_Y) is used as the first program of problem $(C_{Y'})$. Therefore, the same recurrence formulas are used to transform the main part of the principal tableaus when passing to a successive iteration of the Hungarian method. By virtue of (5.4), this applies also to the transformation of column e_0.

We shall illustrate the application of the Hungarian method to linear-programming problems with bilateral restraints by an example which has been previously considered in connection with the simplex method. Maximize the linear form

$$L(X) = x_1 + 2x_2 + 3x_3 + x_4 + 2x_5 + 3x_6 + x_7 - x_8 - x_9 - x_{10} - x_{11}$$

subject to the conditions

$$\begin{aligned} x_1 + x_2 + 2x_3 + 3x_4 + 2x_5 + 3x_6 + x_7 + x_8 &= 7, \\ 2x_1 + 3x_2 + x_3 + x_4 + 3x_5 + 2x_6 + 2x_7 + x_9 &= 8, \\ x_1 + 2x_2 + 3x_3 + 2x_4 + 0.5x_5 + x_6 + x_7 + x_{10} &= 6, \\ 2x_1 + x_2 + 3x_3 + x_4 + 2x_5 + 3x_6 + x_7 + x_{11} &= 7, \\ 0 \leqslant x_j \leqslant 1, \quad j = 1, 2, \ldots, 7, & \\ x_j \geqslant 0, \quad j = 8, 9, 10, 11. & \end{aligned}$$

The process of solution is given in Table 7.7 (principal tableaus) and in Table 7.6 (auxiliary tableau). As the first feasible program of the dual problem we take the vector $Y = (0.8;\ 0;\ 0.6;\ 0)$.

In row Δ of subtableau 1 the parameters Δ_2 and Δ_6 are zero. Hence, $E_Y = \{2,6\}$. The process of solution of the first auxiliary problem is written in principal tableaus 1,1', and 1'' (see Table 7.7) and in subtableau 1 of the auxiliary tableau (see Table 7.6). In row (α, β) of subtableau 1 the elements corresponding to restraint vectors A_2 and A_6 carry the symbol α, since in the first program of auxiliary problem (C_Y) the variables x_2 and x_6 are equal to zero, i.e., the limit on the left of the interval of definition. Other elements of row (α, β) are written according to the sign of the corresponding Δ_j. The entry for A_5, where $\Delta_5 = -0.1$, is the symbol β, and the entry for A_3, where $\Delta_3 = 0.4$, is the symbol α.

The components of quasiprogram X are listed in row $(\alpha, \beta)'$ of subtableau 1 and in column e_0 of principal tableau 1''. The parameter θ_0 of the elementary transformation of kernel Y into kernel Y' of a program of the dual problem is obtained on $j = 3$.

The elements of row Δ of subtableau 2 are computed from recurrence formula (5.3). The set $E_{Y'}$, contains the indices $j = 2$ and 3. The solution of problem (C_Y) is taken as the first program of problem $(C_{Y'})$. The main part of principal tableau 2 is, therefore, computed from the main part of principal tableau 1'' according to the same recurrence formulas as those used to determine the elements e_{ij} in tableau 1'' from the elements of tableau 1'.

The basis components of the optimal program of the problem are written in column e_0 of principal tableau 7. The extrabasis components of the solution are determined from row (α, β) of subtableau 7 of the auxiliary tableau.

§ 6. The Hungarian method and bilateral evaluations

6-1. When solving problem (A) by the Hungarian method we construct a sequence of programs of the dual problem $(\tilde{A})$ which monotonically converge to its solution.

The value of the linear form of the dual problem on these programs monotonically decreases approaching the value $L(X^*)$ of the linear form on the optimal program X^* of the primal problem (A). After each iteration we can, therefore, easily obtain an upper bound for the optimal value of the linear form $L(X)$ of the primal problem (A).

In some problems each quasiprogram can be used to construct a feasible program of the primal problem on which the value of the linear form does not exceed its value on the quasiprogram. In these problems after each iteration we can obtain a program and evaluate the difference between the value of the linear form on this program and its optimal value.

In particular, problems for which a feasible program can easily be determined for any constraint vector $B \geqslant 0$ and where the linear-form coefficients are nonnegative have this property. This can easily be verified from the following considerations. If X' is a quasiprogram of the problem, then $AX' \leqslant B$. Given the structure of the restraint matrix A we can easily find a nonnegative vector X'' satisfying the system of equations

$$AX'' = B - AX'.$$

The vector $X = X' + X''$ is a feasible program of the primal problem and, since the linear-form coefficients are nonnegative,

$$L(X) \geqslant L(X').$$

Examples of problems which have the above property are problems whose matrices contain a complete system of unit vectors and the transportation problem.

6-2. We now consider the revised Hungarian method (which can conveniently be called the m e t h o d o f b i l a t e r a l e v a l u a t i o n s) which, in general, involves covering feasible programs of the primal problem.

Each iteration of this revised Hungarian method is associated with a certain evaluation of the deviation of the program found from the optimum. The problem is considered solved when the evaluation falls within a predetermined range. The revised Hungarian method is often preferred in the solution of various practical problems when a solution ensuring a given deviation from the optimal value of the linear form is acceptable. It is noteworthy, however, that to solve linear-programming problems by the revised Hungarian method we should know, besides the first program Y of the dual problem $(\tilde{A})$, also a first support program X of the primal problem (A).

We outline briefly the general procedure of the revised Hungarian method.

We start with a support program $X = (x_1, x_2, \ldots, x_n)$ of problem (A) and a feasible program $Y = (y_1, y_2, \ldots, y_m)$ of dual problem $(\tilde{A})$. Problem (A) is given in canonical form:

Maximize the linear form

$$L(X) = \sum_{j=1}^{n} c_j x_j \tag{6.1}$$

subject to the conditions

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, 2, \ldots, m, \tag{6.2}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n. \tag{6.3}$$

Further, let $y_1, y_2, \ldots, y_m$ be an arbitrary given set of numbers. Multiplying the i-th restraint in (6.2) by y_i, summing the right- and the left-hand sides of the resulting equalities, and subtracting (6.1) from the sum, we obtain

$$\sum_{i=1}^{m} b_i y_i - L(X) = \sum_{i=1}^{m} \sum_{j=1}^{n} (a_{ij} y_i - c_j) x_j.$$

Setting

$$\Delta_j = \sum_{i=1}^{m} a_{ij} y_i - c_j, \tag{6.4}$$

we find

$$\sum_{j=1}^{n} \Delta_j x_j = \sum_{i=1}^{m} b_i y_i - L(X). \tag{6.5}$$

Thus, the problem of maximizing $L(X)$ under conditions (6.2) and (6.3) can be stated as a problem of minimizing $\sum_{j=1}^{n} \Delta_j x_j$ under the same conditions and with constant $y_i\,(i=1, 2, \ldots, m)$.

Now let $Y=(y_1, \ldots, y_m)$ be a feasible program of problem $(\tilde{A})$, the dual of problem (6.1)–(6.3). Then

$$\Delta_j \geqslant 0, \quad j=1, 2, \ldots, n. \tag{6.6}$$

If X is a program of problem (A), the value of linear form (6.5) can be used to evaluate the deviation of program X from the optimum. Indeed,

$$\sum_{j=1}^{n} c_j x_j \leqslant L(X_{\text{opt}}) \leqslant \sum_{i=1}^{m} b_i y_i.$$

The process of solution can, therefore, be terminated when the difference

$$\sum_{i=1}^{m} b_i y_i - \sum_{j=1}^{n} c_j x_j = \sum_{j=1}^{n} \Delta_j x_j$$

is comparable to the error inherent in measurements of the initial data of the problem.

Let, as before, E_Y be the set of indices j such that $\Delta_j=0$, where Δ_j is computed from (6.4) starting with a program Y of the dual problem. Let E_X be the set of indices of the basis vectors of the support program X. Moreover, we introduce the set $E_{X,Y}$ of all the indices j which belong to E_X or $E_Y\,(E_{X,Y}=E_X \cup E_Y)$.

The pair of programs X and Y is associated with an auxiliary problem $(D_{X,Y})$ which is to minimize of the linear form

$$\sum_{j\in E_{X,Y}} \Delta_j x_j \tag{6.7}$$

subject to the conditions

$$\sum_{j\in E_{X,Y}} A_j x_j = B, \tag{6.8}$$

$$x_j \geqslant 0, \quad j \in E_{X,Y}. \tag{6.9}$$

We solve problem $(D_{X,Y})$ by the second simplex algorithm. As the first program of problem $(D_{X,Y})$ we naturally take the given support program X. Clearly, every program of problem $(D_{X,Y})$ is also a program of problem (A). The components of program X of problem (A) whose indices are not in $E_{X,Y}$ are zero.

Let X^* be the optimal program of problem $(D_{X,Y})$. Relationship (6.5) shows that the value of linear form (6.7) on program X^* can be called the residue of program X^* of problem (A),

$$\varepsilon^* = \sum_{j \in E_{X,Y}} \Delta_j x_j^*. \tag{6.10}$$

The optimality test for programs of problem (A) is stated as follows:

Program X^* is the solution of problem (A) if its residue is zero.

As in the Hungarian method, we introduce the parameters

$$\delta_j^* = \sum_{i=1}^{m} a_{ij}\mu_i^*, \quad j=1, 2, \ldots, n, \tag{6.11}$$

where $M^* = (\mu_1^*, \mu_2^*, \ldots, \mu_m^*)$ is the solution of problem $(\tilde{D}_{X,Y})$, the dual of auxiliary problem $(D_{X,Y})$. The solutions of the auxiliary problem $(D_{X,Y})$ and of its dual problem $(\tilde{D}_{X,Y})$ define a new support program $X' = X^*$ of problem (A) and a feasible program Y' of problem $(\tilde{A})$. The residue of program X' is less than the residue of the first program X.

Program Y is transformed to program Y' of problem $(\tilde{A})$ by means of the elementary transformation

$$Y' = Y - \theta_0 M^*. \tag{6.12}$$

The parameter θ_0 of the elementary transformation is computed from

$$\theta_0 = \min_{\delta_j^* > 0} \frac{\Delta_j}{\delta_j^*}. \tag{6.13}$$

The parameters Δ_j' of one iteration are related with the parameters Δ_j of the preceding iteration by the recurrence formulas

$$\Delta_j' = \Delta_j - \theta_0 \delta_j^*. \tag{6.14}$$

Support program $X' = X^*$ of problem (A) and feasible program Y' of problem $(\tilde{A})$ define a new auxiliary problem $(D_{X',Y'})$. As the first program of problem $(D_{X',Y'})$ we can, clearly, take the optimal program X^* of problem $(D_{X,Y})$.

This process is repeated until a program with zero residue is obtained. Problem (A) is always solvable, since by assumption the process of solution starts with a given support program X of problem (A) and a given feasible program Y of problem $(\tilde{A})$.

6-3. The algorithm of the revised Hungarian method (the method of bilateral evaluations) is different from the computational procedure of the ordinary Hungarian method.

In the Hungarian method, the relative evaluations δ_j of the restraint vectors A_j of the auxiliary problem are defined as

$$\delta_j^{(l)} = \sum_{i=1}^{m} a_{ij}\mu_i^*, \quad j \in E_Y$$

(the coefficients of x_j in the linear form of problem (C_Y) are all zero).

The linear-form coefficients of problem $(D_{X,Y})$ are equal to Δ_j. Therefore in the revised Hungarian method

$$\delta_j^{(l)} = \sum_{i=1}^{m} a_{ij}\mu_i^* - \Delta_j; \quad j \in E_{X,Y}. \tag{6.15}$$

Here l is the number of the last iteration in the solution of the auxiliary problem.

In the Hungarian method

$$\delta_j^* = \delta_j^{(l)}, \quad j \in E_Y,$$

and in the revised Hungarian method

$$\delta_j^* = \delta_j^{(l)} + \Delta_j, \quad j \in E_{X, Y}. \tag{6.16}$$

With $j \in E_Y$, we have $\Delta_j = 0$ and $\delta_j^* = \delta_j^{(l)}$. With $j \in E_{X^*}$, we have $\delta_j^{(l)} = 0$ (optimality test of program X of problem $(D_{X, Y})$ computed by the simplex method). Applying (6.16), we find

$$\delta_j^* = \Delta_j \quad \text{for} \quad j \in E_{X^*},$$

and, consequently (see (6.14)),

$$\Delta_j' = \Delta_j (1 - \theta_0), \quad \text{if} \quad j \in E_{X^*}. \tag{6.17}$$

As the first program of problem $(D_{X', Y'})$ we take the optimal program of problem $(D_{X, Y})$. Therefore, $E_{X'} = E_{X^*}$ and relationship (6.17) holds for all $j \in E_{X'}$.

Some of the Δ_j $(j \in E_{X, Y})$ are positive (otherwise, the residue $\varepsilon^* = 0$). Let, for instance, $\Delta_{j_0} > 0$. Further, since all $\Delta_j' \geqslant 0$, we have

$$(1 - \theta_0) = \frac{\Delta_{j_0}'}{\Delta_{j_0}} \geqslant 0.$$

Clearly, the optimal program of the auxiliary problem will not change if all its linear form coefficients are multiplied by a constant positive number. Formula (6.17) and the relationship $E_{X'} = E_{X^*}$ therefore enable us to replace the solution of problem $(D_{X'}, {}_{Y'})$ with the solution of the reduced auxiliary problem with linear form

$$\sum_{j \in E_{X', Y'}} \Delta_j x_j$$

and the restraints of problem $E_{X', Y'}$.

If A_j is a restraint vector of the two auxiliary problems above, the linear-form coefficients of the two problems corresponding to the vector A_j are equal. This makes it possible to do without various auxiliary operations in each iteration of the revised Hungarian method and to effect the transition from the auxiliary problem to the reduced problem.

The value of the linear form of the auxiliary problem on every support program $X^{(l)}$ differs from the value of the linear form of the corresponding reduced problem on the same program by the factor

$$\nu_l = (1 - \theta_0^{(1)})(1 - \theta_0^{(2)}) \ldots (1 - \theta_0^{(l-1)}),$$

where l is the number of the iteration of the revised Hungarian method. The residue of program $X^{(l)}$ is, therefore, equal to

$$\varepsilon_l^* = \nu_l \mu_0^{*(l)}, \tag{6.18}$$

where $\mu_0^{*(l)}$ is the value of the linear form of the reduced auxiliary problem on program $X^{(l)}$. ε_l^* is conveniently written in the left bottom corner of the principal tableau containing the solution of the l-th reduced auxiliary problem.

The preceding remarks are sufficient to construct a computational

procedure of the revised Hungarian method on the basis of the primal-dual algorithm of the Hungarian method. This is left to the reader (Exercise 11).

We shall now show that as long as we do not have the solution of problem (A), the parameter θ_0 of the elementary transformation falls in the interval $(0, 1)$. In other words,

$$0<\theta_0^{(l)}<1, \tag{6.19}$$

if the residue of program $X^{(l)}$ is positive.

The inequality $\theta_0^{(l)}<1$ for $\varepsilon_l^*>0$ follows from (6.18). On the other hand, if some of δ_j^* are positive,

$$\theta_0 = \min_{\delta_j^*>0} \frac{\Delta_j}{\delta_j^*}>0.$$

To prove the left-hand side of inequality (6.19), it thus suffices to show that some of the parameters δ_j^* are positive.

If $\delta_j^*\leqslant 0$ for all j, problem (A) is unsolvable. Indeed, if $\delta_j^*\leqslant 0$ for all j, the vector $Y(\theta)=Y-\theta M^*$ is a program of problem (A) for any $\theta>0$ (see (6.14)). The value of the linear form of problem $(\tilde{A})$ on program $Y(\theta)$ is equal to

$$\tilde{L}(Y(\theta))=\tilde{L}(Y-\theta M^*)=\tilde{L}(Y)-\theta\sum_{i=1}^{m} b_i\mu_i^*=\tilde{L}(Y)-\theta\cdot\sum_{j\in E_{X,Y}}\Delta_j x_j^*=\tilde{L}(Y)-\theta\varepsilon^*.$$

When $\varepsilon^*>0$, an increase of θ results in unlimited decrease of $\tilde{L}(Y')$. According to Chapter 3, Lemma 1.3 this indicates that problem (A) has no feasible programs. This contradicts the assumption that the process of solution by the revised Hungarian method starts with a given support program of problem (A). Thus, some of δ_j^* are positive. This proves inequality (6.19).

6-4. We shall now show that the revised Hungarian method gives the solution of a linear-programming problem after a finite number of iterations. The proof follows the outline below. We first show that each iteration of the revised Hungarian method either yields a new support program of problem (A) or reduces the number of restraint vectors with positive Δ_j in the successive auxiliary problem. The number of support programs, and also the number of restraint vectors with positive Δ_j, are finite. A method in which a support program has already been examined does not recur or which reduces the number of vectors with positive Δ_j in each iteration is a finite method.

When passing to a successive iteration, two cases may arise in the revised Hungarian method: the parameter θ_0 of the elementary transformation is obtained either on $j\notin E_{X,Y}$ or on $j\in E_{X,Y}$. We shall consider these two possibilities.

1. The parameter θ_0 is obtained on $j\notin E_{X,Y}$. Then

$$\Delta_j'=0 \quad (j\in E_{Y'}),$$

while $\Delta_j>0$. Moreover, we have seen that $\theta_0>0$ if X^* is not a solution of problem (A). It follows from (6.14) that under these conditions $\delta_j^*>0$. In problem $(D_{X',Y'})$ the parameter δ_j^* is an evaluation of the restraint vector A_j. Hence, some of the restraint vectors of problem $(D_{X',Y'})$ have positive evaluations relative to the basis of program X'. In the nondegenerate case (nondegeneracy of problem $(D_{X',Y'})$ which follows from nondegeneracy of problem (A)), the first step of the iteration will yield a program of problem (A) with a lower residue ε^*. Program Y' is not affected in the process of

solution of the auxiliary problem $(D_{X',Y'})$. The decrease of the value of the linear form of the auxiliary problem is, therefore, due to an increase of the linear form of the primal problem.

If $j \notin E_{X,Y}$ the transition to the next successive iteration in the revised Hungarian method involves motion over different support programs of the problem. Using rules which prevent cycling we can extend the result to the degenerate case also.

2. The parameter θ_0 is obtained on $j \in E_{X,Y}$. We shall prove that in this case the transition to the next successive iteration of the revised Hungarian method reduces the number of basis vectors of the auxiliary problem whose Δ_j are positive. This proposition is obvious if the number of basis vectors with positive Δ_j is reduced in the process of solution of problem $(D_{X,Y})$. In this case

$$E_X \supset E_{X^*} = E_{X'}.$$

Now let E_X and $E_{X^*} = E_{X'}$ contain the same number of indices of restraint vectors with $\Delta_j > 0$. The index j on which θ_0 is obtained belongs, by assumption, to the set $E_{X,Y}$. From (6.19),

$$\theta_0 = \frac{\Delta_j}{\delta_j^*} > 0.$$

Hence $\Delta_j > 0$. In other words, $j \in E_X = E_{X'}$.

Thus, on the one hand, $\Delta_j' = 0$ (θ_0 is obtained on j). On the other hand, $j \in E_{X'} = E_{X^*}$. From equality (6.17) and the right-hand side of inequality (6.19) we see that these relationships are consistent only if the residue of program $X^* = X'$ is zero (here X' is regarded as the solution of problem $(D_{X',Y'})$).

Thus, if the solution of the problem has not yet been found, the parameter θ_0 of the elementary transformation can be obtained on $j \in E_{X,Y}$ only if

$$E_X \supset E_{X^*} = E_{X'}.$$

This proves that there are a finite number of steps in the revised Hungarian method.

6-5. We illustrate the application of the revised Hungarian method by the following example.

Example 1. Maximize the linear form

$$L(X) = 13x_1 + 10x_2 + 9x_3 + 6x_4 + 17x_5 + 14x_6 + 15x_7 + 14.5x_8 + 29x_9 + 14x_{10}$$

subject to the conditions

$$1.5x_1 + x_2 + 17x_3 + 12x_4 + 18x_5 + 7.5x_6 + 5x_7 + 4x_8 + 6.5x_9 + 2.5x_{10} = 7,$$
$$9x_1 + 13x_2 + 16x_3 + 5x_4 + 4x_5 + 4x_6 + 6x_7 + 7x_8 + 15x_9 + 8x_{10} = 14.5,$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, 10$$

Solution. The process of solution is given in Table 7.9 (principal tableaus) and Table 7.8 (auxiliary tableau).

As the first support program of the problem we take the program with the basis components $x_2 = 0.656$; $x_3 = 0.373$, and as the feasible program of the dual problem the vector $Y = (2.095;\ 1.095)$. In row Δ of the first subtableau of the auxiliary tableau Δ_1 and Δ_{10} are zero. Therefore, $E_Y = \{1;\ 10\}$. The basis of the first support programs consists of vectors A_2 and A_3. Hence, $E_X = \{2, 3\}$ and $E_{X,Y} = \{1, 2, 3, 10\}$.

The linear form of the first auxiliary problem is equal to

$$\Delta_2 x_2 + \Delta_3 x_3 = 6.333x_2 + 44.14x_3,$$

and the restraint vectors of problem $(D_{X,Y})$ are A_1, A_2, A_3, and A_{10}. The simplex solution of the problem is given in principal tableaus 1 and 1' (see Table 7.9) and in subtableau 1 of the auxiliary tableau (Table 7.8).

The parameter θ_0 of the elementary transformation, defined as the least entry in row θ of subtableau 1, is equal to 0.157. The elementary transformation of program Y of the dual tableau into program Y' yields $E_{Y'} = \{9, 10\}$. The basis of the solution of problem $(D_{X,Y})$ comprises the vectors A_3 and A_{10}. These vectors also constitute the basis of the first program X' of problem $(D_{X',Y'})$. Hence, $E_{X'} = \{3, 10\}$ and $E_{X',Y'} = \{3, 9, 10\}$. The residue of program X' is equal to 9.081.

TABLE 7.8

No.	B	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	Y
1	7	1.5	1	17	12	18	7.5	5	4	6.5	2.5	2.095
2	14,5	9	13	16	5	4	4	6	7	15	8	1.095
3	C	13	10	9	6	17	14	15	14,5	29	14	
1	Δ	×	6.333	44.14	24.62	25.10	6.095	2.048	1.548	1.048	×	
	δ	6.246			—	—	—	—	—	—	**8.241**	
	δ'	—4.828	—		—	—	—	—	—	—		
	δ^*	—4.828	—11.27	44.14	38.40	61.62	22.99	11.50	6.667	6.667		
	θ	—	—	—	0.641	0.407	0.265	0.178	0.232	**0.157**	—	$\theta_0^{(1)}=0.157$
2	$\tilde{\Delta}'$	0.900	2.100	44.14	22.05	18.29	2.946	0.286	0.593	×	×	
	δ	—	—		—	—	—	—	—		—4.239	
	δ^*	—10.524	—20.32	44.14	43.12	71.33	25.29	10.524	4.239		—4.239	
	θ	—	—	—	0.511	0.256	0.116	**0.0272**	0.140	—	—	$\theta_0^{(2)}=0.0272$
3	$\tilde{\Delta}''$	1.220	2.727	44.14	21.46	16.80	2.321	×	0.491	×	0.119	

TABLE 7.9 (1-3)

	No.	σ	B_x	e_0	e_1	e_2	A_k	θ	No. of tableau
←	1	6.333	A_2	0.656	−0.0780	0.0829	0.468	**1.401**	1
	2	44.14	A_3	0.373	0.0634	−0.0049	0.120	3.122	
	3	μ_1	20.63	20.63	2.305	0.310	8.241	—	
⇄	1		A_{10}	1.401	−0.167	0.177	1.573	**0.891**	1
	2	44.14	A_3	0.206	0.0833	−0.0260	0.151	1.362	
	3	μ_1'	9.081	9.081	3.679	−1.150	6.667	—	
→	1		A_9	0.891	−0.106	0.113	0.146	6.114	2
←	2	44.14	A_3	0.0712	0.0993	−0.0430	0.238	**2.986**	
	3	μ_2	2.650	3.143	4.385	−1.900	10.52	—	
	1		A_9	0.847	−0.167	0.139			3
→	2		A_7	0.299	0.417	−0.181			
	3	μ_3	0	0	0	0		—	

According to 6-3 it is advisable to substitute a reduced auxiliary problem for problem $(D_{X',Y'})$. The restraint vectors of the reduced problem belong to $E_{X',Y'}$, and its linear form is

$$\sum_{j\in E_{X',Y'}} \tilde{\Delta}'_j x_j = \sum_{j\in E_{X',Y'}} \frac{\Delta'_j}{1-\theta_0} x_j.$$

This eliminates the necessity of making any changes whatsoever in tableau 1' before starting the solution of the next auxiliary problem.

In the first row of subtableau 1, therefore, we write the values

$$\tilde{\Delta}'_j = \frac{\Delta'_j}{1-\theta_0} = \frac{\Delta_j - \theta_0\delta^*_j}{1-\theta_0},$$

where, as we have seen, $\tilde{\Delta}'_j = \Delta_j = \delta^*_j$ for $j\in E_{X'}$.

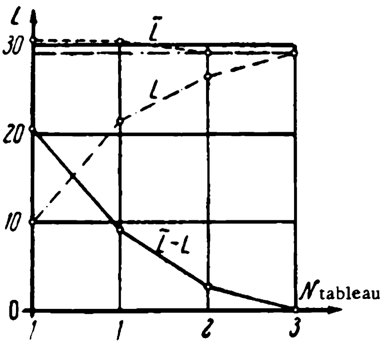


FIGURE 7.6

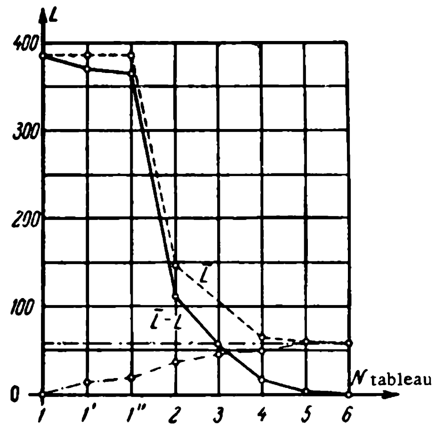


FIGURE 7.7

The simplex solution of the auxiliary problem is given in principal tableaus 1' and 2 (see Table 7.9) and in subtableau 2 of the auxiliary tableau (see Table 7.8). The basis components of the solution of the reduced auxiliary problem are written in column e_0 of principal tableau 2. The $(m+1)$-th (third) entry in column e_0 is the value of the linear form of the reduced auxiliary problem on its optimal program. To obtain the linear form of problem $(D_{X',Y'})$ on this program (the residue of the corresponding program of the primal problem), we multiply $e_{m+1,0}$ by $(1-\theta_0)$. The residues are written in the $(m+1)$-th position of column B_X (to the left of $e_{m+1,0}$).

The solution of the reduced auxiliary problem is a support program of the primal problem. Its basis components are

$$x''_3 = 0.0712, \quad x''_9 = 0.891.$$

Hence, $E_{X''} = \{3, 9\}$. Elementary transformation with $\theta'_0 = 0.0272$ yields feasible program Y'' of the dual problem. The elements in the upper row $\tilde{\Delta}''$ of the subtableau corresponding to the successive iteration of the revised Hungarian method are computed from the parameters of the preceding iteration using the formula

$$\tilde{\Delta}''_j = \frac{\tilde{\Delta}'_j - \theta'_0\delta^{*\prime}_j}{1-\theta'_0}.$$

The parameter $\tilde{\Delta}''_j = 0$ for $j=7$ and 9. Hence, $E_{Y''} = \{7, 9\}$ and $E_{X'',Y''} = \{3, 7, 9\}$. The linear form of the reduced auxiliary problem with restraint vectors A_3, A_7, A_9 on the optimal program of this problem is

$$\tilde{\Delta}''_3 x^*_3 + \tilde{\Delta}''_7 x^*_7 + \tilde{\Delta}''_9 x^*_9 = 44,\ 14\cdot 0 + 0\cdot 0.847 + 0\cdot 0.299 = 0.$$

The support program X'' of the primal problem with basis components $x''_7 = 0.847$ and $x''_9 = 0.299$ has a zero residue. This program thus solves the problem.

Figure 7.6 shows the graph of the program residue in successive iterations of the revised Hungarian method. This figure also shows the variation of the linear forms of the primal and the dual problems on the corresponding programs. Successive plotting of these graphs in the process of solution makes it possible to discontinue the computation as soon as the residue is comparable to the error of measurements of the initial data.

TABLE 7.10

No.	B	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	Y
1	12	−6	9	3		−2	−1	1					
2	5		−4	3	−3	1	−1		1				
3	20	2	8	−5	6	−8	4			1			
4	10	−1	−3	−4	−8		4				1		5
5	24	5	1	2	4	9	5					1	14
6	C	3	−1	8	2	−1	9						
1	Δ	62	×	×	14	127	81	×	×	×	5	14	
	δ	—	−1	**8**	—	—	—						
	δ'	—	**9.667**		—	—	—		−2.667				
	δ''	—			—	—	—	−0.744	−1.923				
	δ^*	69.46			21.77	125.56	92.67	−0.744	−1.923		5	14	
	θ	0.893	—	—	**0.643**	1.011	0.874	—	—	—	1	1	$\theta_0^{(1)}=0.643$

TABLE 7.10 (continued)

No.	B	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	Y
2	$\tilde{\Delta}$	48.55	×	×	×	129.59	59.98	1.340	3.465	×	5	14	
	δ	—	−94.33			—	—	—	—				
	δ^*	113	−94.33			147.33	92.67	−8	5.333		5	14	
	θ	**0.430**	—	—	—	0.880	0.647	—	0.650	—	1	1	$\theta_0^{(2)}=0.430$
3	$\tilde{\Delta}$	×	71.07	×	×	116.22	35.35	8.377	2.058	×	5	14	
	δ		—			—	—	—	—			—	
	δ^*		54.12			52.06	55	5.294	−3.529		5	7.353	
	θ	—	1.313	—	—	2.232	**0.643**	1.582	—	—	1	1.904	$\theta_0^{(3)}=0.643$
4	$\tilde{\Delta}$	×	101.57	×	×	231.62	×	13.92	12.11	×	5	25.96	
	δ	−165	—			—		—	—			—	
	δ^*	−165	270.88			−87.06		24.71	−16.47		5	−2.353	
	θ	—	**0.375**	—	—	—	—	0.563	—	—	1	—	$\theta_0^{(4)}=0.375$
5	$\tilde{\Delta}$	98.98	×	×	×	422.79	×	7.452	29.25	×	5	42.94	
	δ	—			−62.51	—		—	—			—	
	δ^*	−39.98			−62.51	−24.55		3.869	4.367		5	−2.353	
	θ	—	—	—	—	—	—	1.93	6.698	—	**1**	—	$\theta_0^{(5)}=1$
6	$\tilde{\Delta}$												

TABLE 7.11 (1–6)

	No.	σ	B	e_0	e_1	e_2	e_3	e_4	e_5	A_k	θ	No. of tableau
	1		A_7	12	1					3	4	
←	2		A_8	5		1				3	**1.667**	
	3		A_9	20			1			−5	—	1
	4	5	A_{10}	10				1		−4	—	
	5	14	A_{11}	24					1	2	12	
	6	μ_1	386	386				5	14	8	—	
←	1		A_7	7	1	−1				13	**0.538**	
→	2		A_3	1.667		0.333				−1.333	—	
	3		A_9	28.33		1.667	1			1.333	21.25	1′
	4	5	A_{10}	16.67		1.333		1		−8.333	—	
	5	14	A_{11}	20.67		−0.667			1	3.667	5.636	
	6	μ_1'	372.67	372.67		−2.667		5	14	9.667	—	

TABLE 7.11 (continued)

	No.	σ	B	e_0	e_1	e_2	e_3	e_4	e_5	A_k	θ	No. of tableau
$\rightleftarrows$	1		A_2	0.538	0.0769	−0.0769				0.231	**2.333**	1″
	2		A_3	2.385	0.1026	0.231				−0.692	—	
	3		A_9	27.62	−0.1026	1.769	1			0.692	3.444	
	4	5	A_{10}	21.15	0.641	0.692		1		10.08	—	
	5	14	A_{11}	18.69	−0.282	−0.385			1	5.154	3.627	
	6	μ_1''	367.46	367.46	−0.744	−1.923		5	14	21.77	—	
→	1		A_4	2.333	0.333	−0.333				−2	—	2
	2		A_3	4	0.333					−2	—	
	3		A_9	26	−0.333	2	1			4	6.5	
	4	5	A_{10}	44.67	4	−2.667		1		−25	—	
←	5	14	A_{11}	6.667	−2	1.333			1	17	**0.392**	
	6	μ_2	113.02	316.67	−8	5.333		5	14	113	—	

TABLE 7.11 (continued)

	No.	σ	B	e_0	e_1	e_2	e_3	e_4	e_5	A_k	θ	No. of tableau
	1		A_4	3.118	0.0980	−0.176			0.118	0.667	4.676	3
	2		A_8	4.784	0.0980	0.157			0.118	0.333	14.35	
	3		A_9	24.43	0.137	1.686	1		−0.235	1	24.43	
	4	5	A_{10}	54.47	1.059	−0.706		1	1.471	11	4.952	
⇄	5		A_1	0.392	−0.118	0.0784			0.0588	0.333	**1.176**	
	6	μ_3	55.43	272.35	5.294	−3.529		5	7.353	55	—	
←	1		A_4	2.333	0.333	−0.333				4.333	**0.538**	4
	2		A_3	4.392	0.216	0.0784			0.0588	1.686	2.605	
	3		A_9	23.25	0.686	1.451	1		−0.412	6.196	3.753	
	4	5	A_{10}	41.53	4.941	−3.294		1	−0.471	54.18	0.767	
→	5		A_6	1.176	−0.353	0.235			0.176	−3.941	—	
	6	μ_4	15.10	207.65	24.71	−16.47		5	−2.353	270.88	—	

TABLE 7.11 (continued)

	No.	σ	B	e_0	e_1	e_2	e_3	e_4	e_5	A_k	q	No. of tableau
→	1		A_2	0.538	0.0769	−0.0769					—	5
	2		A_3	3.484	0.0860	0.208			0.0588		—	
	3		A_9	19.92	0.0136	1.928	1		−0.412		—	
←	4	5	A_{10}	12.36	0.774	0.873		1	−0.471	1	12.36	
	5		A_6	3.299	−0.0498	−0.0679			0.176		—	
	6	μ_5	2.809	61.79	3.869	4.367		5	−2.353		—	
	1		A_2	0.538	0.0769	−0.0769						6
	2		A_3	3.484	0.0860	0.208			0.0588			
	3		A_9	19.92	0.0136	1.928	1		−0.412			
→	4		A_{10}	12.36	0.774	0.873		1	−0.471			
	5		A_6	3.299	−0.0498	−0.0679			0.176			
	6	μ_6	0	0	0	0	0	0	0		—	

Example 2. Maximize the linear form

$$L = 3x_1 - x_2 + 8x_3 + 2x_4 - x_5 + 9x_6$$

subject to the conditions

$$\begin{aligned}
-6x_1 + 9x_2 + 3x_3 - 2x_5 - x_6 &\leq 12,\\
-4x_2 + 3x_3 - 3x_4 + x_5 - x_6 &\leq 5,\\
2x_1 + 8x_2 - 5x_3 + 6x_4 - 8x_5 + 4x_6 &\leq 20,\\
-x_1 - 3x_2 - 4x_3 - 8x_4 + 4x_6 &\leq 10,\\
5x_1 + x_2 + 2x_3 + 4x_4 + 9x_5 + 5x_6 &\leq 24,\\
x_j \geq 0, \quad j = 1, 2, \ldots, 6.
\end{aligned}$$

Solution. This problem has already been considered as an illustration to other linear-programming methods. Table 7.11 (principal tableaus) and Table 7.10 (auxiliary tableau) show the process of its solution by the revised Hungarian method. The system of additional unit vectors was taken as the basis of the first support program. The first program of the dual prbolem was taken as $Y = (0, 0, 0, 5, 14)$.

Figure 7.7 shows the graphs of $L(X)$, $\tilde{L}(Y)$, and the program residue in successive iterations.

6-6. In conclusion of this chapter we wish to mention some advantages of the Hungarian method which distinguish it favorably from other linear-programming methods.

To compute the optimal program by the Hungarian method it is not necessary to determine first a support program of either the primal or the dual problem. The process of solution can start with any feasible program of the dual problem, and the solution of a problem of bilateral restraints may start with an arbitrary m-dimensional vector Y.

Given a support program of the primal problem and a feasible program of the dual problem, we may use the revised Hungarian method (the method of bilateral evaluations) to obtain, in some cases, an approximate solution of the problem with an error not exceeding a predetermined value, after appreciably fewer operations than those required to obtain the exact components of the optimal program.

EXERCISES TO CHAPTER 7

1. Using the Hungarian method, maximize the linear form

$$L = 2x_1 - 4x_2 + x_3 + x_4 - 2x_5$$

subject to the conditions

$$\begin{aligned}
x_1 - 3x_2 + 2x_3 \qquad - x_5 &= 1,\\
-2x_2 + 3x_3 - x_4 + x_5 &= 5,\\
2x_1 + x_2 + 2x_3 + 5x_4 + x_5 &= 10,\\
x_j \geq 0, \quad j = 1, 2, 3, 4.
\end{aligned}$$

2. Solve the problem in Exercise 1 imposing the additional restraints: $x_j \leq 2$, $j = 1, 2, 3, 4, 5$.

3. Solve the problem in Exercise 1 by the revised Hungarian method (the method of bilateral evaluations) starting with a support program $X = (0, 0, 41/29, 31/29, 53/29)$ and a feasible program of the dual problem $Y = (0, 0, 1)$.

4. Starting with a feasible program $Y = (0, 0, 1)$ of the dual problem, find the parameters a in the problem which is to maximize linear form

$$L = \sum_{j=1}^{4} x_j$$

subject to the conditions

$$\begin{aligned}
-x_1 + ax_2 + \frac{7}{5}x_3 + ax_4 &= 3,\\
-x_1 + \frac{7}{5}x_2 + ax_3 + \frac{7}{5}x_4 &= 7,\\
3x_1 + 2x_2 + 2x_3 + x_4 &= 5,\\
x_j \geq 0, \quad j = 1, 2, 3, 4,
\end{aligned}$$

for which cases (a), (b), and (c) of the Hungarian method apply.

5. Show that the convergence of the Hungarian method is not impaired if the artificial unit vectors eliminated from the basis are not included in successive bases.

6. Let the vector $M^* = (\mu_1^*, \ldots, \mu_m^*)$ be defined by (4.16). Show that M^* is the kernel of the solution of the dual problem (4.8), (4.11), (4.12).

7. Show that formula (4.5) applies to a problem of minimizing linear form (4.1) subject to conditions (4.2), (4.3) if γ_j are defined as

$$\gamma_j = \begin{cases} \alpha_j & \text{for} \quad \Delta_j < 0, \\ \beta_j & \text{for} \quad \Delta_j > 0. \end{cases}$$

8. Draw a block-diagram of the primal-dual algorithm of the Hungarian method for problems with bilateral restraints.

9. Let problem (M) (see 5-3) be associated with a linear-programming problem. Let $\tilde{X}^* = (x_0, x_1, \ldots, x_n)$ be the solution of problem (M). If $j=0$ belongs to the set $E_{Y^{(s)}}$ where $Y^{(s)}$ is a feasible program of the dual problem corresponding to the last iteration of the Hungarian method, then $X^* = (x_1, \ldots, x_n)$ is the solution of the primal problem. If $j=0$ does not belong to $E_{Y^{(s)}}$, the primal problem is unsolvable. Prove.

10. Prove that if the process of solution of problem (M) terminates in case (b) (problem (M) is unsolvable), the restraints of the primal problem are inconsistent.

11. Using the remarks given in 6-3 and the examples described in 6-5, compile a computational procedure and draw a block-diagram of the revised Hungarian method (the method of bilateral evaluations).

12. Give a geometrical interpretation of the revised Hungarian method (the method of bilateral evaluations) in the $(m+1)$-dimensional space.

Chapter 8

FINITE METHODS OF LINEAR PROGRAMMING

The various methods of linear programming can be divided into two classes. The methods of the first class (the so-called finite method) ensure a solution of a problem after a finite number of steps. The methods of the second class (iterative methods) involve an infinite number of iterations and, generally speaking, yield an approximate solution of the problem. The quality of the approximation depends to a large extent on the number of iterations carried out. Most of the iterative methods are actually numerical methods for solving rectangular games stated in terms of linear programming.

In Chapters 4 to 7 we studied in detail the qualitative and computational aspects of three principal methods based on essentially different premises. These were the simplex method, the dual (simplex) method, and the Hungarian method. In this chapter we shall consider some problems which, since they can be solved by all the finite methods, were not discussed when dealing with each method separately.

Until now our discussion of the methods and of the corresponding algorithms was presented mainly with reference to the linear-programming problem in canonical form. Uniformity of notations simplifies the analysis of the methods and the description of the computational procedures. This, however, should not be interpreted to indicate that each applied problem must first be reduced to canonical form. Retaining the original form of the problem we often greatly reduce the bulkiness of the computations. The problem with bilateral restraints given in Chapter 7 serves as a good example of this. In § 1 the simplex method is extended to linear-programming problems in arbitrary form. Following the suggestions given in this section, the reader will learn to adapt the other finite methods and the corresponding algorithms to problems written in arbitrary form.

In § 2 we discuss some means of perfecting the simplex and the dual simplex methods which will simplify the corresponding computational procedures. Modification of the methods discussed in this section enables us to use the memory storage space of computers, which is the bottleneck in all modern computing machines, more economically.

In § 3 the finite methods are classified. The various features outlining a unified approach to all the finite methods of linear programming are considered.

§ 1. Finite methods and linear-programming problems in arbitrary form

1-1. In this section we shall deal with the general linear-programming problem given in arbitrary form. The restraints of this problem comprise

both equalities and inequalities; the nonnegativity restraints are imposed only on some of the problem variables. Any of the finite linear-programming methods can be suited to problems in arbitrary form. This adaptation should be made setting out from the general outline of the method described for the problem in canonical form and drawing upon the geometrical interpretation of the method. We shall deal here with only one method, the simplex procedure.

Consider the linear-programming problem of maximization of the linear form

$$\sum_{j=1}^{n} c_j x_j \tag{1.1}$$

subject to the conditions

$$\sum_{j=1}^{n} a_{ij} x_j \begin{cases} = b_i, & i = 1, 2, \ldots, p; \\ \leqslant b_i, & i = p+1, p+2, \ldots, m; \end{cases} \qquad \begin{matrix} (1.2) \\ (1.3) \end{matrix}$$

$$x_j \geqslant 0, \quad j = q+1, \quad q+2, \ldots, n. \tag{1.4}$$

Equations (1.2) should, naturally, be assumed to be linearly independent; otherwise some equations could have been omitted. Moreover, we shall assume that the system of vectors

$$A_j = (a_{1j}, a_{2j}, \ldots, a_{mj})^T, \qquad j = 1, 2, \ldots, q$$

is linearly independent. This condition does not detract from the generality of our arguments: it can always be fulfilled by reducing the number of variables which do not satisfy restraint (1.4) (see Chapter 2, 4-1). The last assumption ensures the existence of support programs for problem (1.1)-(1.4).

The definition of a support program given in Chapter 2, 4-2 can be restated with reference to problem (1.1)-(1.4) as follows.

A feasible program $X = (x_1, x_2, \ldots, x_n)$ of problem (1.1)-(1.4) is called a s u p p o r t p r o g r a m, if there exists a regular square submatrix

$$A_X = \left\| \begin{matrix} a_{i_1 j_1} & a_{i_1 j_2} & \ldots & a_{i_1 j_s} \\ a_{i_2 j_1} & a_{i_2 j_2} & \ldots & a_{i_2 j_s} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ a_{i_s j_1} & a_{i_s j_2} & \ldots & a_{i_s j_s} \end{matrix} \right\|$$

of the matrix of coefficients $A = \| a_{ij} \|_{m, n}$, such that

(a) $i_\alpha = \alpha$ for $\alpha = 1, 2, \ldots, p$; $j_\beta = \beta$ for $\beta = 1, 2, \ldots, q$;

(b) if $q < j$ and $x_j > 0$, then $j = j_\beta$ for some $\beta \geqslant q+1$;

(c) $\sum_{j=1}^{n} a_{i_\alpha j} x_j = b_{i_\alpha}$ for $\alpha = 1, 2, \ldots, s$.

The rows of the matrix A_X are constructed of restraints (1.2) and some of the restraints (1.3), all such inequalities being satisfied by program X as strict equalities. Among the columns of A_X there are all the columns of the matrix of coefficients A which correspond to the components x_j which are either not restrained by the nonnegativity requirement or else are positive on program X. The submatrix A_X generated by the support program X will be called t h e b a s i s o f t h i s p r o g r a m. According to the general definition given in Chapter 2, 4-7, the support program X of problem (1.1)-(1.4) with basis A_X is said to be n o n d e g e n e r a t e, if

$$x_{j_\beta} > 0, \qquad \beta = q+1, \ldots, s;$$

$$\sum_{j=1}^{n} a_{ij} x_j < b_i, \qquad i \neq i_\alpha, \quad \alpha = 1, 2, \ldots, s.$$

Proof of the equivalence of the above definitions of a support program and of nondegeneracy of a program of problem (1.1)–(1.4) with the corresponding definitions given in Chapter 2 is left to the reader (Exercises 1 and 2).

1-2. The simplex method will be described here with reference to the nondegenerate case – all the support programs of problem(1.1)–(1.4) are nondegenerate.

Let a support program X of problem (1.1)–(1.4) be known and let its basis A_X be formed by the elements occupying, in matrix A, the intersections of the rows and the columns with the numbers $i_1, i_2, \ldots, i_s$ and $j_1, j_2, \ldots, j_s$, respectively. We begin to solve the problem by testing program X for optimality. To this end, we determine the solution $(\lambda_1, \lambda_2, \ldots, \lambda_s)$ of the system

$$\sum_{\alpha=1}^{s} a_{i_\alpha j_\beta}\lambda_\alpha = c_{j_\beta}, \qquad \beta = 1, 2, \ldots, s, \tag{1.5}$$

and compute the parameters

$$\Delta_j = \sum_{\alpha=1}^{s} a_{i_\alpha j}\lambda_\alpha - c_j, \qquad j = 1, 2, \ldots, n.$$

If

$$\lambda_\alpha \geqslant 0, \qquad \alpha = p+1, \ldots, s \tag{1.6}$$

and

$$\Delta_j \geqslant 0 \quad \text{for} \quad j = 1, 2, \ldots, n, \tag{1.7}$$

then, according to the optimality criterion for problem (1.1)–(1.4) (Chapter 3, Theorem 5.2), program X is optimal (case (a)). If, however, at least one of relationships (1.6), (1.7) does not apply, program X is transformed by elementary transformation which either establishes unsolvability of the problem (case (b)) or yields an improved support program X' (case (c)).

We recall the geometrical interpretation of elementary transformation in the simplex method (see Chapter 4, 2-2). A support program corresponds to a vertex of the polyhedral set of the problem. We choose one of the hyperplanes passing through this vertex and satisfying the inequality restraints and consider the edge of the polyhedral set contained in the intersection of the remaining hyperplanes. The elementary transformation associated with the chosen hyperplane is a translation along the edge indicated. If the problem has been reduced to canonical form, elementary transformation of its support program is invariably associated with one of the coordinate hyperplanes. In the general case, elementary transformation may be generated either by a hyperplane of the form $x_j = 0$ $(j = q+1, \ldots, n)$, or by a hyperplane

$$\sum_{j=1}^{n} a_{lj}x_j = b_l, \qquad l = p+1, p+2, \ldots, m,$$

associated with the corresponding restraint in (1.3). Therefore, when analyzing elementary transformations of a program of problem (1.1)–(1.4), it is advisable to distinguish between two cases.

1-3. We consider the first case. The elementary transformation is defined by the coordinate hyperplane $x_k = 0$ $(k \neq j_\beta, \beta = 1, 2, \ldots, s)$ (elementary transformation of the first kind).

Under this transformation, the support program X is transformed into

$$X(\theta) = X - \theta H = (x_1(\theta), x_2(\theta), \ldots, x_n(\theta)),$$

where the vector $H=(h_1, h_2, \ldots, h_n)$ is determined by the relationship

$$h_j=\begin{cases} x_{\beta k}, & \text{if} \quad j=j_\beta \quad \beta=1,\ 2, \ldots, s; \\ -1, & \text{if} \quad j=k; \\ 0 & \text{in other cases.} \end{cases} \tag{1.8}$$

Here $x_{\beta k}$ are computed from the equations

$$\sum_{\beta=1}^{s} a_{i_\alpha j_\beta} x_{\beta k}=a_{i_\alpha k}, \qquad \alpha=1,\ 2, \ldots, s. \tag{1.9}$$

It follows from (1.8) and (1.9) that

$$\sum_{j=1}^{n} a_{i_\alpha j} x_j(\theta)=b_{i_\alpha} \quad \text{for} \quad \alpha=1,\ 2, \ldots, s$$

for any θ.

We write out the conditions imposed on θ under which $X(\theta)$ is a feasible program, i. e., satisfies the remaining restraints in (1.3) and constraints (1.4). Since

$$x_k(\theta)=x_k+\theta,$$

the parameter θ must be nonnegative. Insofar as the other restraints of system (1.4) must also be satisfied, we have

$$x_{j_\beta}-\theta x_{\beta k} \geqslant 0, \qquad \beta=q+1, \ldots, s.$$

Thus, for all the components of the vector $X(\theta)$, beginning with the $(q+1)$-th, to be nonnegative, it is necessary and sufficient that

$$0 \leqslant \theta \leqslant \theta_0', \tag{1.10}$$

where

$$\theta_0'=\min_{\substack{x_{\beta k}>0 \\ \beta>q}} \frac{x_{j_\beta}}{x_{\beta k}}. \tag{1.11}$$

If $x_{\beta k} \leqslant 0$ for $\beta=q+1, \ldots, s$ the vector $X(\theta)$ has nonnegative components $x_j(\theta)$ $(j=q+1, \ldots, n)$ for any nonnegative θ. In this case we take $\theta_0'=\infty$.

We define

$$\left.\begin{aligned} \Delta^{(i)} &= b_i-\sum_{j=1}^{n} a_{ij} x_j, \\ \delta_k^{(i)} &= \sum_{\beta=1}^{s} a_{ij_\beta} x_{\beta k}-a_{ik}, \\ i &= 1,\ 2, \ldots, m. \end{aligned}\right\} \tag{1.12}$$

Applying (1.8) and (1.12), we have

$$\Delta^{(i)}(\theta)=b_i-\sum_{j=1}^{n} a_{ij} x_j(\theta)=\Delta^{(i)}+\theta\delta_k^{(i)}.$$

Restraints (1.3), to be satisfied by the vector $X(\theta)$, are therefore rewritten in the form

$$\Delta^{(i)}+\theta\delta_k^{(i)} \geqslant 0, \qquad i=p+1, \ldots, m. \tag{1.13}$$

Solving inequalities (1.13) for $\theta \geqslant 0$ we obtain

$$\theta \leqslant \theta_0'', \tag{1.14}$$

where

$$\theta_0''=\min_{\delta_k^{(i)}<0}\left(-\frac{\Delta^{(i)}}{\delta_k^{(i)}}\right). \tag{1.15}$$

If $\delta^{(i)} \geqslant 0$ for all i, we take $\theta_0''=\infty$.

Inequalities (1.1) and (1.14) define the set of θ on which $X(\theta)$ is a feasible program of problem (1.1)–(1.4):

$$0 \leqslant \theta \leqslant \theta_0,$$

where

$$\theta_0 = \min\{\theta_0', \theta_0''\}, \tag{1.16}$$

where θ_0' and θ_0'' are determined from (1.11) and (1.15), respectively.

We compute the increase in the value of the linear form (1.1) caused by transformation from X to $X(\theta)$. We have

$$\sum_{j=1}^{n} c_j x_j(\theta) = \sum_{j=1}^{n} c_j x_j - \theta\left(\sum_{\beta=1}^{s} c_{j_\beta} x_{\beta k} - c_k\right).$$

Applying systems (1.5) and (1.9), we have

$$\sum_{\beta=1}^{s} c_{j_\beta} x_{\beta k} = \sum_{\beta=1}^{s}\left(\sum_{\alpha=1}^{s} a_{i_\alpha j_\beta}\lambda_\alpha\right) x_{\beta k} = \sum_{\alpha=1}^{s}\left(\sum_{\beta=1}^{s} a_{i_\alpha j_\beta} x_{\beta k}\right)\lambda_\alpha = \sum_{\alpha=1}^{k} a_{i_\alpha k}\lambda_\alpha.$$

Hence,

$$L(X(\theta)) = \sum_{j=1}^{n} c_j x_j(\theta) = L(X) - \theta\Delta_k. \tag{1.17}$$

Consider the elementary transformation associated with the hyperplane

$$\sum_{j=1}^{n} a_{i_t j} x_j = b_{i_t} \qquad (t = p+1,\ p+2,\ \ldots,\ s)$$

(elementary transformation of the second kind).

Under this transformation program X is transformed into

$$X(\theta) = (x_1(\theta),\ \ldots,\ x_n(\theta)) = X - \theta H,$$

where the vector $H = (h_1, h_2, \ldots, h_n)$ is determined from

$$h_j = \begin{cases} e_{\beta t}, & \text{if } \ j = j_\beta,\ \beta = 1, 2, \ldots, s, \\ 0 & \text{in other cases.} \end{cases} \tag{1.18}$$

The $e_{\beta t}$ satisfy the equations

$$\sum_{\beta=1}^{s} a_{i_\alpha j_\beta} e_{\beta t} = \begin{cases} 0 & \text{for } \alpha \neq t, \\ 1 & \text{for } \alpha = t. \end{cases} \tag{1.19}$$

Let us find the exact limits of θ between which $X(\theta)$ is a feasible program of problem (1.1)–(1.4). From (1.19),

$$\sum_{j=1}^{n} a_{i_\alpha j} x_j(\theta) = \sum_{j=1}^{n} a_{i_\alpha j} x_j - \theta \sum_{\beta=1}^{s} a_{i_\alpha j_\beta} e_{\beta t} = \begin{cases} b_{i_\alpha}, & \text{if } \alpha \neq t, \\ b_{i_\alpha} - \theta, & \text{if } \alpha = t. \end{cases} \tag{1.20}$$

Therefore, restraints (1.2), (1.3) are satisfied for $i = i_\alpha$, $\alpha = 1, 2, \ldots, s$, if and only if θ is nonnegative.

We define

$$\bar{\delta}_i^{(t)} = \sum_{j=1}^{n} a_{ij} h_j = \sum_{\beta=1}^{s} a_{ij_\beta} e_{\beta t}, \qquad i = 1, 2, \ldots, m.$$

Then

$$\Delta^{(i)}(\theta) = b_i - \sum_{j=1}^{n} a_{ij} x_j(\theta) = \Delta^{(i)} + \theta\bar{\delta}_i^{(t)}.$$

Since θ is nonnegative, applying restraints (1.3) to be satisfied by the vector $X(\theta)$ we obtain

$$\theta \leqslant \theta_0'',$$

where

$$\theta_0'' = \min_{\delta_t^{(i)} < 0} \left(-\frac{\Delta^{(i)}}{\bar{\delta}_t^{(i)}} \right). \tag{1.21}$$

If $\bar{\delta}_t^{(i)} \geqslant 0$ for all i, then $\theta_0'' = \infty$.

From the definition of $X(\theta)$,

$$x_j(\theta) = 0 \quad \text{for } j \neq j_\beta, \quad \beta = 1, 2, \ldots, s.$$

Hence, restraints (1.4) are fulfilled if and only if

$$x_{j_\beta}(\theta) = x_{j_\beta} - \theta e_{\beta t} \geqslant 0, \qquad \beta = q+1, \ldots, s,$$

or, since θ is nonnegative,

$$\theta \leqslant \theta_0',$$

where

$$\theta_0' = \min_{\substack{e_{\beta t} > 0 \\ \beta > q}} \frac{x_{j_\beta}}{e_{\beta t}}. \tag{1.22}$$

If $e_{\beta t} \leqslant 0$ for $\beta = q+1, \ldots, s$, we take $\theta_0' = \infty$. The required limits of θ are thus specified by

$$0 \leqslant \theta \leqslant \theta_0,$$

where

$$\theta_0 = \min(\theta_0', \theta_0''), \tag{1.23}$$

where θ_0' and θ_0'' are computed from (1.22) and (1.21), respectively.

Under a given elementary transformation of the second kind, the linear form is written as

$$L(X(\theta)) = L(X) - \theta \sum_{j=1}^{n} c_j h_j.$$

Applying (1.18), (1.5), and (1.19), successively, we obtain

$$\sum_{j=1}^{n} c_j h_j = \sum_{\beta=1}^{s} c_{j_\beta} e_{\beta t} = \sum_{\beta=1}^{s} \left(\sum_{\alpha=1}^{s} a_{i_\alpha j_\beta} \lambda_\alpha \right) e_{\beta t} = \sum_{\alpha=1}^{s} \left(\sum_{\beta=1}^{s} a_{i_\alpha j_\beta} e_{\beta t} \right) \lambda_\alpha = \lambda_t.$$

Hence

$$L(X(\theta)) = L(X) - \theta \lambda_t. \tag{1.24}$$

1-4. We now analyze cases (b) and (c). Let some of the $\lambda_\alpha (\alpha = p+1, \ldots, s)$ and $\Delta_j (j = 1, 2, \ldots, n)$ be negative. We choose one of these negative numbers (e.g., the one of greatest absolute value). Now the direction of the solution process is determined by whether one of the Δ_j or one of the λ_α has been chosen. We shall deal separately with each of the two cases.

1. Let $\Delta_k < 0$ be chosen. In this case, we apply elementary transformation of the first kind defined by hyperplane $x_k = 0$. We compute θ_0 from (1.11), (1.15), and (1.16). It is possible that $\theta_0 = \infty$. This is equivalent to the following set of restraints:

$$\left. \begin{array}{ll} x_{\beta k} \leqslant 0, & \beta = q+1, q+2, \ldots, s, \\ \delta_k^{(i)} \geqslant 0, & i = 1, 2, \ldots, m. \end{array} \right\} \tag{1.25}$$

If $\theta_0 = \infty$, $X(\theta)$ is a feasible program of problem (1.1)-(1.4) for any $\theta \geqslant 0$. But according to (1.17)

$$\lim_{\theta \to \infty} L(X(\theta)) = \infty.$$

Thus, if conditions (1.25) are satisfied (case (b)), the problem in question is unsolvable.

Now assume that conditions (1.25) are not satisfied (case (c)) and, consequently, $\theta_0<\infty$. Let

$$X'=X(\theta_0).$$

We shall verify that X' is a support program of the problem. First let $\theta_0=\theta_0'$, or equivalently,

$$\theta_0=\frac{x_{jr}}{x_{rk}}, \quad x_{rk}>0,$$

for some $r\geqslant q+1$.

Since $x_{rk}\neq 0$, it follows from Chapter 4, Theorem 2.1 that matrix $A_{X'}$, which is obtained from the basis A_X of program X by replacing the r-th column by the column $(a_{i_1k}, a_{i_2k}, \ldots, a_{i_sk})^T$, is regular.

Program X' is thus a support program and its basis is a square matrix of order s.

Let us consider the possibility:

$$\theta_0=\theta_0''.$$

Here

$$\theta_0=-\frac{\Delta^{(l)}}{\delta_k^{(l)}}, \quad \delta_k^{(l)}<0$$

for some $l\neq i_\alpha$, $\alpha=1, 2, \ldots, s$. Obviously,

$$\sum_{j=1}^{n} a_{lj}x_j'=b_l;$$

$$x_j'=0, \quad \text{if } j\neq j_\beta,\ \beta=1, 2, \ldots, s, \text{and } j\neq k.$$

Therefore, if we verify that the matrix

$$A_{X'}=\begin{Vmatrix} a_{i_1j_1} & a_{i_1j_2} & \ldots & a_{i_1j_s} & a_{i_1k} \\ a_{i_2j_1} & a_{i_2j_2} & \ldots & b_{i_2j_s} & a_{i_2k} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{i_sj_1} & a_{i_sj_2} & \ldots & a_{i_sj_s} & a_{i_sk} \\ a_{lj_1} & a_{lj_2} & \ldots & a_{lj_s} & a_{lk} \end{Vmatrix},$$

is regular, we shall have proved that X' is a support program.

The proof will be carried out by reductio ad absurdum. If the columns of matrix $A_{X'}$ are linearly dependent, its last column is a linear combination of the first s columns since these columns are a priori linearly independent. According to (1.9), the coefficients of the linear combination are $x_{\beta k}$. However, by assumption,

$$\delta_k^{(l)}=\sum_{\beta=1}^{s} a_{lj_\beta}x_{\beta k}-a_{lk}\neq 0$$

Hence, the assumption of linear dependence of the columns of $A_{X'}$ is false; matrix $A_{X'}$ is regular.

Program X' is thus a support program, and its basis $A_{X'}$ is a matrix of the order $s+1$ which is obtained by adding a row $(a_{lj_1}, a_{lj_2}, \ldots, a_{lj_s}, a_{lk})$ and a column $(a_{i_1k}, a_{i_2k}, \ldots, a_{i_sk}, a_{lk})^T$ to the matrix A_X.

By assumption X is a nondegenerate support program of problem (1.1)–(1.4).

Hence, $x_{j_r}>0$, $\Delta^{(l)}>0$, and, consequently, $\theta_0>0$. Referring to (1.17), we find that

$$L(X')=L(X)-\theta_0\Delta_k>L(X).$$

2. Let $\lambda_t<0$ be chosen. In this case, the program is improved by taking an elementary transformation of the second kind — that generated by the hyperplane

$$\sum_{j=1}^{n} a_{i_t j} x_j = b_{i_t}\,.$$

We compute θ_0 from (1.21)–(1.23). If $\theta_0=\infty$ or, equivalently, if

$$\left.\begin{aligned} h_{j_\beta} &= e_{\beta t} \leqslant 0 \qquad \text{for } \beta=q+1,\ \ldots,\ s; \\ \bar\delta_t^{(i)} &= \sum_{\beta=1}^{s} a_{ij_\beta} e_{\beta t} \geqslant 0 \text{ for } i=1,2,\ \ldots,\ m, \end{aligned}\right\} \tag{1.26}$$

then $X(\theta)$ is a feasible program of problem (1.1)–(1.4) for any $\theta\geqslant 0$. Equality (1.26) shows that the problem in question is unsolvable (case (b)).

If conditions (1.26) are not satisfied, $\theta_0<\infty$ and the program $X'=X(\theta_0)$ can be determined (case (c)).

As in the previous case, with the elementary transformation of the first kind, X' proves to be a support program. To verify this we must again consider the two possibilities arising when computing θ_0.

Let $\theta_0=\theta_0'$. Then

$$\theta_0=\frac{x_{j_r}}{e_{rt}} \qquad (e_{rt}>0)$$

for some $r\geqslant q+1$. In this case

$$x_j'=0,\quad j\neq j_\beta,\qquad \beta=1,\ 2,\ \ldots,\ r-1,\ r+1,\ \ldots,\ s;$$

$$\sum_{j=1}^{n} a_{i_\alpha j} x_j' = b_{i_\alpha},\qquad \alpha=1,\ 2,\ \ldots,\ t-1,\ t+1,\ \ldots,\ s.$$

We now show that the square matrix of $(s-1)$-th order

$$A_{X'}=\begin{Vmatrix} a_{i_1j_1} & \ldots & a_{i_1j_{r-1}} & a_{i_1j_{r+1}} & \ldots & a_{i_1j_s} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{i_{t-1}j_1} & \ldots & a_{i_{t-1}j_{r-1}} & a_{i_{t-1}j_{r+1}} & \ldots & a_{i_{t-1}j_s} \\ a_{i_{t+1}j_1} & \ldots & a_{i_{t+1}j_{r-1}} & a_{i_{t+1}j_{r+1}} & \ldots & a_{i_{t+1}j_s} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{i_sj_1} & \ldots & a_{i_sj_{r-1}} & a_{i_sj_{r+1}} & \ldots & a_{i_sj_s} \end{Vmatrix}$$

is regular. Indeed, the determinant of the matrix $A_{X'}$ is equal, up to the sign, to the cofactor of the element $a_{i_t j_r}$ of the matrix A_X. Therefore

$$|A_{X'}|=\pm e_{rt}\,|A_X|\neq 0,$$

since $e_{rt}\neq 0$. Thus X' is a support program with the basis $A_{X'}$ which constitutes a matrix of order $(s-1)$ obtained from A_X when the t-th row and the r-th column are crossed out.

We now admit the possibility: $\theta_0=\theta_0''$. Then

$$\theta_0=-\frac{\Delta^{(l)}}{\bar\delta_t^{(l)}} \qquad (\bar\delta_t^{(l)}<0)$$

for some $l\neq i_\alpha\ (\alpha=1,\ 2,\ \ldots,\ s)$. Obviously

$$x_j'=0 \ \text{ for } j\neq j_\beta,\ \beta=1,\ 2,\ \ldots,\ s;$$

$$\sum_{j=1}^{n} a_{ij} x_j' = b_i \text{ for } i=i_\alpha,\ \ \alpha=1,\ 2,\ \ldots, t-1,\ t+1,\ \ldots,\ s,\ i=l.$$

We now show that $A_{X'}$ obtained from A_X when the t-th row is replaced by $(a_{lj_1}, a_{lj_2}, \ldots, a_{lj_s})$ is a regular matrix.

According to (1.19), the vector $(e_{1t},\ e_{2t},\ \ldots,\ e_{st})^T$ is the t-th column of the

matrix A_X^{-1}. The coefficient of the t-th row when the vector $(a_{tj_1}, a_{tj_2}, \ldots, a_{tj_s})$ is expressed in terms of the rows of the matrix A_X is, therefore, equal to

$$\bar{\delta}_t^{(t)} = \sum_{\beta=1}^{s} a_{tj_\beta} e_{\beta t} \neq 0.$$

Further, applying Chapter 4, Theorem 2.1, to the vector $(a_{tj_1}, a_{tj_2}, \ldots, a_{tj_s})$ and to the rows of A_X, we conclude that $A_{X'}$ is a regular matrix. Therefore, in this case, X is a support program with the basis $A_{X'}$, which is a matrix of order s.

Since program X is assumed to be nondegenerate, $\theta_0 > 0$. Hence, according to (1.24),

$$L(X') = L(X) - \lambda_t \theta > L(X).$$

1-5. In short, given a support program X of problem (1.1)-(1.4), we either establish its optimality (case (a)), or prove unsolvability of the problem (case (b)), or else improve program X' increasing the linear form (1.1) (case (c)). The transformation from program X to program X' is an iteration (step) of the method. Since the increase of the linear form (1.1) in each iteration is monotonic and the number of support programs of problem (1.1)-(1.3) is finite, the method is called finite.

We emphasize that until now we have been dealing with the case of nondegenerate problems. Only this assumption ensures that θ_0 be positive and, consequently, the monotonic increase of linear form (1.1) in each iteration.

The parameter θ_0 required for improving the given support program X is computed as a minimum of some set of numbers. When elementary transformation of the first kind is applied, this set is defined by

$$\begin{gathered} \frac{x_{l\beta}}{x_{\beta k}} \ (\beta = q+1,\ q+2,\ \ldots,\ s,\ x_{\beta k} > 0), \\ -\frac{\Delta^{(i)}}{\delta_k^{(i)}} \ (i \neq i_\alpha \text{ for } \alpha = 1,\ 2,\ \ldots,\ s;\ \delta_k^{(i)} < 0). \end{gathered} \tag{1.27}$$

For elementary transformations of the second kind, the set is defined by

$$\begin{gathered} \frac{x_{l\beta}}{e_{\beta t}} \ (\beta = q+1,\ q+2,\ \ldots,\ s;\ e_{\beta t} > 0), \\ -\frac{\Delta^{(i)}}{\bar{\delta}_t^{(i)}} \ (i \neq i_\alpha \text{ for } \alpha = 1,\ 2,\ \ldots,\ s;\ \bar{\delta}_t^{(i)} < 0). \end{gathered} \tag{1.28}$$

It can be easily verified that in the nondegenerate case the sets (1.27) or (1.28) each have one least element (see Exercise 3). This element, as has been shown in 1-4, uniquely defines the basis of the new support program X'. In the degenerate case, the minimum may be obtained simultaneously on several elements of the corresponding set (1.27) or (1.28). To determine the basis of the new support program, one of the several elements equal to θ_0 must be chosen. If the element is chosen arbitrarily cycling may arise, i. e., periodic return to the same basis. (This is known from experience.) However, since cycling is fairly rare, in applied problems one of the least numbers in sets (1.27) or (1.28) may, nevertheless, be chosen arbitrarily. We shall not dwell here on the rule which guarantees against cycling in the solution of any problem of type (1.1)-(1.4). It suffices to note that this rule can be formed with the aid of the ε-procedure used repeatedly in the previous chapters.

1-6. Some remarks on the simplex algorithm as applied to problem (1.1)-(1.4) are now necessary. It follows from (1.19) that the numbers

$e_{1t}, e_{2t}, \ldots, e_{st}$ constitute the t-th column of the inverse of the matrix

$$A_X = \|a_{i_\alpha j_\beta}\|_s ,$$

which is the basis of the support program X. Hence,

$$\|e_{\beta\alpha}\|_s = \|a_{i_\alpha j_\beta}\|_s^{-1}.$$

We shall show that applying the inverse matrix $\|e_{\beta\alpha}\|_s$ of the basis A_x we can easily determine all the parameters required in an iteration for improving the program X.

1. The parameters λ_α, $\alpha = 1, 2, \ldots, s$ required for testing the program X for optimality satisfy (1.5) or, in matrix form,

$$\Lambda_X A_X = C_X. \tag{1.29}$$

Here $\Lambda_X = (\lambda_1, \ldots, \lambda_s)$, $A_X = \|a_{i_\alpha j_\beta}\|_s$, $C_X = (c_{j_1}, \ldots, c_{j_s})$. Multiplying both sides of (1.29) on the right by

$$A_X^{-1} = \|e_{\beta\alpha}\|_s,$$

we obtain

$$\Lambda_X = C_X \|e_{\beta\alpha}\|_s,$$

or, equivalently,

$$\lambda_\alpha = \sum_{\beta=1}^{s} c_{j_\beta} e_{\beta\alpha} , \qquad \alpha = 1, 2, \ldots, s. \tag{1.30}$$

2. If elementary transformation of the first kind is used for program improvement, the numbers $x_{\beta k}$, $\beta = 1, 2, \ldots, s$ constituting the solution of system (1.9) must be determined. In matrix form, system (1.9) is written as

$$A_X X^{(k)} = A_k^{(X)},$$

where $X^{(k)} = (x_{1k}, x_{2k}, \ldots, x_{sk})^T$, $A_k^{(X)} = (a_{i_1k}, a_{i_2k}, \ldots, a_{i_sk})^T$. Hence,

$$X^{(k)} = A_X^{-1} A_k^{(X)}.$$

Expanding the last equality we have

$$x_{\beta k} = \sum_{\alpha=1}^{s} e_{\beta\alpha} a_{i_\alpha k}, \qquad \beta = 1, 2, \ldots, t. \tag{1.31}$$

3. When elementary transformation of the second kind is applied, the analog of the vector $X^{(k)}$ is the corresponding column (the t-th column) of the matrix $\|e_{\beta\alpha}\|_s$.

Thus, in order to carry out a single iteration of the simplex method without solving systems of equations, it is sufficient to know the inverse of the basis of the program to be improved.

1-7. We shall now derive recurrence formulas for the elements of the inverse matrices of two successive (i. e., obtained one from the other in a single iteration) bases.

We recall a proposition used repeatedly in previous chapters (see Chapter 5, 1-3). Consider a regular square matrix $\|d_{ij}\|_\rho$ of order ρ. Let $\|\tilde{d}_{ij}\|_\rho = \|d_{ij}\|_\rho^{-1}$. If in $\|d_{ij}\|_\rho$ the j_0-th column is replaced by the vector $D_0 = (d_{10}, d_{20}, \ldots, d_{\rho 0})^T$ the inverse $\|\tilde{d}'_{ij}\|_\rho$ of the new matrix is determined from

$$\tilde{d}'_{ij} = \begin{cases} \tilde{d}_{ij} - \dfrac{\tilde{d}_{j_0 j} d_{i0}}{\tilde{d}_{j_0 0}}, & i \neq j_0, \\ \dfrac{\tilde{d}_{j_0 j}}{\tilde{d}_{j_0 0}}, & i = j_0, \end{cases} \tag{1.32}$$

$$i=1,\ 2,\ \ldots,\ \varrho,\ j=1,\ 2,\ \ldots,\ \varrho.$$

Here $\tilde{d}_{j0},\ j=1, 2,\ \ldots,\ \varrho$ are the coefficients of the expansion of the vector D_0 in terms of the columns $D_j=(d_{1j},\ d_{2j},\ \ldots,\ d_{\rho j})^T$ of the matrix $\|d_{ij}\|_\rho$:

$$\tilde{d}_{j0}=\sum_{\gamma=1}^{\varrho}\tilde{d}_{j\gamma}d_{\gamma 0}. \tag{1.33}$$

Now substitute the vector $D^{(0)}=(d_{01},\ d_{02},\ \ldots,\ d_{0\rho})$ in place of the i-th row in the matrix $\|d_{ij}\|_\rho$. The corresponding inverse is again denoted by $\|\tilde{d}'_{ij}\|_\rho$. Since $\|\tilde{d}_{ij}\|_\varrho^T$ is the inverse of the matrix $\|d_{ij}\|_\varrho^T$ we apply (1.32) and (1.33) and obtain recurrence formulas for the elements of $\|\tilde{d}'_{ij}\|_\rho$ and $\|\tilde{d}_{ij}\|_\rho$:

$$\tilde{d}'_{ij}=\begin{cases}\tilde{d}_{ij}-\dfrac{\tilde{d}_{ii_0}}{\tilde{d}_{0i_0}}\tilde{d}_{0j}, & j\neq i_0,\\[2mm] \dfrac{\tilde{d}_{ii_0}}{\tilde{d}_{0i_0}}, & j=i_0,\end{cases} \tag{1.34}$$

$$i=1,\ 2,\ \ldots,\ \varrho,\quad j=1,\ 2,\ \ldots,\ \varrho,$$

where

$$\tilde{d}_{0i}=\sum_{\gamma=1}^{\varrho}\tilde{d}_{\gamma i}d_{0\gamma},\ i=1,\ 2,\ \ldots,\varrho. \tag{1.35}$$

We now deal with the case when the new matrix is obtained from $\|d_{ij}\|_\rho$ by adding one row and one column to the latter. Schematically, the new matrix $\|d'_{ij}\|_{\rho+1}$ can be represented as follows:

$$\|d'_{ij}\|_{\rho+1}=\left\|\begin{array}{c|c} d_{ij} & d_{i0}\\ \hline d_{0j} & d\end{array}\right\|\begin{array}{c}\uparrow\\ \varrho\\ \downarrow\end{array}\qquad (d\neq 0). \tag{1.36}$$

$$\leftarrow\varrho\rightarrow$$

Using this notation, we introduce the matrices

$$\|d_{ij}\|_{\varrho+1}^{(1)}=\left\|\begin{array}{c|c} d_{ij} & 0\\ \hline 0 & 1\end{array}\right\|,$$

$$\|d_{ij}\|_{\varrho+1}^{(2)}=\left\|\begin{array}{c|c} d_{ij} & 0\\ \hline d_{0j} & d\end{array}\right\|.$$

Multiplication of the corresponding matrices gives, immediately,

$$\|\tilde{d}_{ij}\|_{\varrho+1}^{(1)}=\left\|\begin{array}{c|c} \tilde{d}_{ij} & 0\\ \hline 0 & 1\end{array}\right\|$$

which is the inverse of $\|d_{ij}\|_{\varrho+1}^{(1)}$. The matrix $\|d_{ij}\|_{\varrho+1}^{(2)}$ differs from $\|d_{ij}\|_{\varrho+1}^{(1)}$ only in the last row. Applying (1.34), we obtain

$$\|\tilde{d}_{ij}\|_{\varrho+1}^{(2)}=\left\|\begin{array}{c|c} \tilde{d}_{ij} & 0\\ \hline -\tilde{d}_{0j}/d & 1/d\end{array}\right\|$$

for the inverse of $\|d_{ij}\|_{\varrho+1}^{(2)}$. Here the $\tilde{d}_{0j}$ are determined from (1.35).

The matrix $\|d'_{ij}\|_{\rho+1}$ differs from $\|d_{ij}\|_{\varrho+1}^{(2)}$ only in the last column. Therefore, the elements of the matrix

$$\|\tilde{d}'_{ij}\|_{\rho+1}-\|d'_{ij}\|_{\varrho+1}^{-1}$$

can be determined from the recurrence formulas (1.32) when applied to matrix $\|\tilde{d}_{ij}\|_{\varrho+1}^{(2)}$, which is known.

We now compute the coefficients $d_i^{(2)}$ in the expansion of the vector $\|d_{ij}\|_{\varrho+1}^{(2)}$ $(d_{10},\ d_{20},\ \ldots,\ d_{\rho 0},\ d)^T$ in terms of columns of $\|\tilde{d}_{ij}\|_{\varrho+1}^{(2)}$ applying the explicit

2154

expressions for the elements of the matrix

$$d_i^{(2)}=\begin{cases}\sum_{\gamma=1}^{\varrho}\tilde{d}_{i\gamma}d_{\gamma 0}=\tilde{d}_{i0}, & i=1,\ 2,\ \ldots,\ \varrho,\\ 1-\frac{1}{d}\sum_{\gamma=1}^{\varrho}\tilde{d}_{0\gamma}d_{\gamma 0}, & i=\varrho+1.\end{cases} \tag{1.37}$$

From formulas (1.32) and equalities (1.37), we obtain

$$\tilde{d}_{ij}=\begin{cases}\tilde{d}_{ij}+\frac{\tilde{d}_{0j}d_{i0}}{\tilde{d}_0} & \text{for } i,\ j=1,\ 2,\ \ldots,\ \varrho;\\ -\frac{\tilde{d}_{0j}}{\tilde{d}_0} & \text{for } i=\varrho+1,\ j=1,\ 2,\ \ldots,\ \varrho;\\ -\frac{\tilde{d}_{i0}}{\tilde{d}_0} & \text{for } i=1,\ 2,\ \ldots,\ \varrho,\ j=\varrho+1;\\ \frac{1}{\tilde{d}_0} & \text{for } i=j=\varrho+1.\end{cases} \tag{1.38}$$

The parameters $\tilde{d}_{0j}$ and $\tilde{d}_{i0}$ are determined from (1.35) and (1.33) respectively, and

$$\tilde{d}_0=d-\sum_{\gamma=1}^{\varrho}\tilde{d}_{0\gamma}d_{\gamma 0}=d-\sum_{\gamma=1}^{\varrho}d_{0\gamma}\tilde{d}_{\gamma 0}. \tag{1.39}$$

Finally, consider the last case when one row (the i_0-th) and one column (the j_0-th) are crossed out from $\|d_{ij}\|_\varrho$. We thus obtain the matrix $\|d'_{ij}\|_{\varrho-1}$ of order $\varrho-1$. Let the inverse of the new matrix $\|d'_{ij}\|_{\varrho-1}$ be denoted by $\|\tilde{d}'_{ij}\|_{\varrho-1}$. We introduce two auxiliary matrices of order ϱ: $\|d_{ij}\|_\varrho^{(1)}$, which is obtained from $\|d_{ij}\|_\varrho$ when the i_0-th row is replaced by the j_0-th unit row; and $\|d_{ij}\|_\varrho^{(2)}$ which is obtained from $\|d_{ij}\|_\varrho^{(1)}$ when the j_0-th column is replaced by the i_0-th unit vector.

The relationship between the inverses of the matrices $\|d_{ij}\|_\varrho$, $\|d_{ij}\|_\varrho^{(1)}$ and $\|d_{ij}\|_\varrho^{(1)}$, $\|d_{ij}\|_\varrho^{(2)}$ can be established proceeding from (1.34) and (1.32), respectively.

If from the inverse of the matrix $\|d_{ij}\|_\varrho^{(1)}$ we cross out the i_0-th column and the j_0-th row, we obtain the desired matrix $\|\tilde{d}'_{ij}\|^{-1}$.

The above considerations lead to the following recurrence formulas for the elements of the matrices $\|\tilde{d}_{ij}\|_\varrho$ and $\|\tilde{d}_{ij}\|_{\varrho-1}$:

$$\tilde{d}'_{ij}=\begin{cases}\tilde{d}''_{ij}, & i<j_0, \quad j<i_0,\\ \tilde{d}''_{i+1,j}, & i\geqslant j_0, \quad j<i_0,\\ \tilde{d}''_{i,\ j+1}, & i<j_0, \quad j\geqslant i_0,\\ \tilde{d}''_{i+1,\ j+1}, & i\geqslant j_0, \quad j\geqslant i_0,\end{cases} \tag{1.40}$$

where

$$\tilde{d}''_{ij}=\tilde{d}_{ij}-\frac{\tilde{d}_{i_0 j}\tilde{d}_{i i_0}}{\tilde{d}_{j i_0}},\quad i\neq j_0,\ j\neq i_0. \tag{1.41}$$

Proof of formulas (1.40) and (1.41) is left to the reader (see Exercise 4).

1-8. In the process of improving a program there are four ways of modifying the basis. These are determined by the kind of the elementary transformation chosen and by whether the parameter θ_0 is equal to θ'_0 or to θ''_0. The preceding formulas enable us to derive recurrence formulas relating the elements of the corresponding inverse matrices for each of these cases.

1. Elementary transformation of the first kind generated by the

hyperplane $x_k=0$ is applied, and

$$\theta_0=\theta_0'=\frac{x_{jr}}{x_{rk}}.$$

The basis $A_{X'}$ of the new support program X' is obtained from the basis A_X of program X when the r-th column is replaced by $(a_{i_1k}, a_{i_2k}, \ldots, a_{i_sk})^T$.

Let the inverses of A_X and $A_{X'}$ be denoted by $\|e_{ij}\|_s$ and $\|e'_{ij}\|_s$, respectively.

Applying (1.32), (1 33), we obtain

$$e'_{ij}=\begin{cases} e_{ij}-\dfrac{e_{rj}x_{ik}}{x_{rk}}, & i\neq r,\\ \dfrac{e_{rj}}{x_{rk}}, & i=r,\end{cases} \qquad (1.42)$$
$$i,\ j=1,\ 2,\ \ldots,\ s.$$

Here x_{ik} — the coefficients in the expansion of the vector $A_k^{(X)}$ in terms of the columns of basis A_X — are computed from formula (1.31).

2. The same elementary transformation as in the previous case is applied, but now

$$\theta_0=\theta_0''=-\frac{\Delta^{(l)}}{\delta_k^{(l)}}.$$

In this case the basis $A_{X'}$ of program X' is obtained by adding a row $(a_{lj_1}, a_{lj_2}, \ldots, a_{lj_s}, a_{lk})$ and a column $(a_{i_1k}, a_{i_2k}, \ldots, a_{i_sk}, a_{lk})^T$ to basis A_X. Observe that the coefficients in the expansion of $(a_{lj_1}, a_{lj_2}, \ldots, a_{lj_s})$ in terms of the rows of basis A_X are determined from

$$\bar{\delta}_\alpha^{(l)}=\sum_{\beta=1}^{s} a_{lj_\beta}e_{\beta\alpha}, \qquad \alpha=1,2,\ \ldots,\ s. \qquad (1.43)$$

Applying (1.38), we obtain the following expression for the elements of $\|e'_{ij}\|_{s+1}$, the inverse of matrix $A_{X'}$:

$$e'_{ij}=\begin{cases} e_{ij}+\dfrac{\bar{\delta}_j^{(l)}x_{ik}}{e_0} & \text{for } i,\ j=1,\ 2,\ \ldots,\ s,\\ -\dfrac{\bar{\delta}_j^{(l)}}{e_0} & \text{for } i=s+1, j=1,\ 2,\ \ldots,\ s,\\ -\dfrac{x_{ik}}{e_0} & \text{for } i=1,\ 2,\ \ldots,\ s, j=s+1,\\ \dfrac{1}{e_0} & \text{for } i=j=s+1.\end{cases} \qquad (1.44)$$

Here

$$e_0=a_{lk}-\sum_{\gamma=1}^{s}\bar{\delta}_\gamma^{(l)}a_{i_\gamma k}=a_{lk}-\sum_{\gamma=1}^{s}a_{lj_\gamma}x_{\gamma k}.$$

3. Elementary transformation of the second kind generated by the hyperplane

$$\sum_{j=1}^{n}a_{i_tj}x_j=b_{i_t}$$

is applied, and

$$\theta_0=\theta_0'=\frac{x_{jr}}{e_{rt}}.$$

Under these assumptions, the basis $A_{X'}$ of program X' is obtained from A_X when the t-th row and the r-th column are crossed out. Therefore, to obtain the necessary recurrence formulas between the elements of $\|e_{ij}\|_s$ and $\|e'_{ij}\|_{s-1}$, the inverse of the basis $A_{X'}$, we may apply formulas (1.40) and

(1.41). Thus,

$$e'_{ij}=\begin{cases} e_{ij}-\dfrac{e_{rj}e_{it}}{e_{rt}}, & \text{if } i<r,\ j<t; \\ e_{i+1,\,j}-\dfrac{e_{rj}e_{i+1,\,t}}{e_{rt}}, & \text{if } i\geqslant r,\ j<t; \\ e_{i,\,j+1}-\dfrac{e_{r,\,j+1}e_{it}}{e_{rt}}, & \text{if } i<r,\ j\geqslant t; \\ e_{i+1,\,j+1}-\dfrac{e_{r,\,j+1}e_{i+1,\,t}}{e_{rt}}, & \text{if } i\geqslant r,\ j\geqslant t; \end{cases} \tag{1.45}$$
$$(i,\ j=1,\ 2,\ \ldots,\ s-1).$$

4. The last possibility arises when the elementary transformation of case (c) is applied, but

$$\theta_0=\theta''_0=-\frac{\Delta^{(l)}}{\delta_l^{(l)}}.$$

The basis $A_{X'}$ of program X' is obtained from A_X by substituting the row $(a_{lj_1},\ a_{lj_2},\ \ldots,\ a_{lj_s})$ for the t-th row. Applying formulas (1.34), (1.35), and remembering (1.43), we obtain the following expression for the elements of the matrix $\|e'_{ij}\|_s$, the inverse of $A_{X'}$:

$$e'_{ij}=\begin{cases} e_{ij}-\dfrac{e_{it}}{\delta_t^{(l)}}\,\delta_j^{(l)}, & j\neq t, \\ \dfrac{e_{it}}{\delta_t^{(l)}}, & j=t, \end{cases} \tag{1.46}$$
$$i,\ j=1,\ 2,\ \ldots,\ s,$$

The recurrence formulas (1.42), (1.44)–(1.46), and also the preceding relationships (1.30), (1.31) serve as the basis for the simplex algorithm. The reader is advised to compile this algorithm (see Exercise 5). It suffices to note that the principal part of the algorithm tableaus is filled with the elements of the matrix $\|e_{ij}\|_s$ and has, thus, the order $s\times s$, where

$$\max(p,\ q)\leqslant s\leqslant \min(m,\ n)=\varphi_1.$$

1-9. To conclude, we make the following remarks. To reduce problem (1.1)–(1.4) to canonical form, we must introduce $m-p$ new nonnegative variables (equal to the number of inequalities in system (1.3)) and eliminate q old variables not restrained by the requirement of nonnegativity. The resulting problem has $n+m-(p+q)$ nonnegative variables with $m-q$ equality restraints. Therefore, to solve the reduced problem by the second simplex algorithm, we have to deal with inverse matrices of order $\varphi_2=m-q$. We mentioned at the end of 1-8 that the order of the inverse matrices used in solving the problem by the procedure described in this section is at most

$$\varphi_1=\min(m,\ n).$$

When these methods are applied to universal digital computers, the inverse matrix to be transformed in each iteration is generally stored in the working memory. The dimensions of this matrix are, therefore, a principal parameter necessary to decide whether the problem "fits" into the machine. In manual computations it is also advisable to keep the order of the matrix as low as possible.

Hence, whether to reduce problem (1.1)–(1.4) to canonical form or to solve it by the method described in this section should be decided on the basis of a comparison between two numbers

$$\varphi_1=\min(m,\ n)$$

and

$$\varphi_2 = m - q.$$

If $\varphi_2 \leqslant \varphi_1$, reduction of the problem to canonical form is justified. If, however,

$$\varphi_2 \geqslant \varphi_1 = n,$$

it is expedient to apply the modified simplex method.

Observe that for $p \geqslant 1$ and $q \geqslant 1$ the dimension of problem (1.1)–(1.4) can be reduced by eliminating

$$\tau = \min(p, q)$$

variables (of the first q variables) and omitting τ equations from (1.2). With these transformations we obtain a new problem of type (1.1)–(1.4), where at least one of the parameters (p, q) is a priori equal to zero.

However, when reducing the number of variables the restraint matrix A is changed. Therefore, it may occur that the restraint matrix A' of the equivalent problem will be less convenient for analysis than the matrix A; for instance, the matrix A may have far more zero elements than the matrix A'. In such cases the transformation for reduction will, obviously, complicate computations; slight reduction in the dimension of the problem will, to a large extent, increase the laboriousness of a single iteration. If the matrix A has no useful features and $p, q \geqslant 1$, it is advisable to eliminate part of the variables in problem (1.1)–(1.4).

These remarks emphasize further the preliminary analysis of problem restraints.

1-10. There are some classes of linear-programming problems which are best solved in their original form. We shall give some of them.

Consider the general linear-programming problem in n nonnegative variables with m equality restraints $(m < n)$. The dual problem contains variables not restrained by the nonnegativity requirement, and has n inequality restraints. Therefore, for the dual problem

$$\varphi_1 = \min(m, n) = m; \quad \varphi_2 = n - m.$$

If $m < n - m$ it is advisable to solve the dual problem without modification. In Chapter 6, § 8 we showed that the dual simplex method is an application of the simplex method to the dual problem. It is obvious that the dual problem is then solved according to the procedure described in this section. Since none of the dual-problem variables is restrained by the requirement of nonnegativity, the solution is based only on elementary transformations of the second kind with $\theta_0 = \theta_0''$ throughout.

Consequently, only case (d) is realized when improving the program of the dual problem: the bases of the successive support programs differ only in one row (or, in terms of the primal problem, in one restraint vector).

As another example, consider the statement of the general linear-programming problem proposed by L. V. Kantorovich /64/ (in application of the industrial-planning problem).

An industrial plant, or a group of plants must release m_2 different output products in a predetermined proportion (in a given assortment) specified by the numbers $\lambda_1, \lambda_2, \ldots, \lambda_{m_2}$. In the process m_1 production factors are used, whose resources comprise $b_1, b_2, \ldots, b_{m_1}$ units, respectviely. By assumption, $b_i \geqslant 0\ (i = 1, 2, \ldots, m_1)$. If $b_i = 0$, we take it that the i-th factor is a semifinished product of the given plant.

There is a certain number (n) of different production modes. The j-th production mode is characterized by the vector

$$\bar{A}_j=(a_{1j},\ a_{2j},\ \ldots,\ a_{mj})^T,$$

where $m=m_1+m_2$. The first m_1 components of the vector A_j determine the consumption of the corresponding production factors by the j-th production modes in unit time (unit intensity). The other m_2 components give the quantities of the corresponding output products produced in unit time by the given production mode. The principal purpose of the plant is to arrive at maximum possible output observing the proportions specified and drawing upon the resources available.

A production program is a set $X=(x_1,\ x_2,\ \ldots,\ x_n)$ of utilization intensities of the various production modes.

The problem of determining the optimal production program can be stated mathematically as follows.

Determine the vector $X=(x_1,\ x_2,\ \ldots,\ x_n,\ V)$ satisfying the restraints

$$\sum_{j=1}^{n} a_{m_1+i,\,j}x_j \geqslant \lambda_i V, \qquad i=1,\ 2,\ \ldots,\ m_2; \tag{1.47}$$

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i, \qquad i=1,\ 2,\ \ldots,\ m_1; \tag{1.48}$$

$$x_j \geqslant 0, \qquad j=1,\ 2,\ \ldots,\ n, \tag{1.49}$$

and having a maximum possible $(n+1)$-th component

$$L(X)=V. \tag{1.50}$$

Here V is the number of assemblies of the output products which can be released under the program $X=(x_1,\ x_2,\ \ldots,\ x_n)$, where by assembly we mean a set of these products in the respective amounts $\lambda_1,\ \lambda_2,\ \ldots,\ \lambda_{m_2}$.

Problem (1.47)–(1.50) has $n+1$ variables, n of which are assumed nonnegative, and $m=m_1+m_2$ inequality restraints. In this case

$$\varphi_1=\min(n+1,\ m_1+m_2),$$
$$\varphi_2=m_1+m_2-1.$$

In general $n \ll m_1+m_2$ and, consequently, $\varphi_1<\varphi_2$. This indicates that it is advisable to solve problem (1.47)–(1.50) in its original form, i.e., as originally stated.

In the next section, where we consider a modification of the simplex method based on simultaneous introduction of several restraint vectors into the basis, we shall again encounter the necessity of applying the technique discussed here.

Finally, consider the class of problems of best uniform approximation of functions defined on a finite system of points by linear combinations of a fixed set of functions. Application of the simplex method as described in this section to problems of this class leads to the method suggested by S. I. Zukhovitskii /60/.

In this section we dealt only with the simplex method. This, however, does not mean that other general methods of linear programming are only applicable to problems in canonical form. Each general linear-programming method can also be adapted to problems of type (1.1)–(1.4) in a manner similar to that shown in this section for the simplex method.

§ 2. Modification of finite methods

2-1. Consider the general linear-programming problem in canonical form.

The simplex method entails a successive examination of adjoining support programs of the problem whose bases differ only in one vector. The vector to be introduced into the basis is chosen from among the restraint vectors with negative relative evaluations. There are usually several restraint vectors with this property, and each is suitable for inclusion in the basis. However, following the recommendations of the simplex method, we choose only one such vector. The question is whether the simplex method can be modified so that several suitable restraint vectors can be introduced simultaneously into the basis. The same question can be posed regarding the dual simplex method.

When the dual simplex method is used, the solution procedure entails examination of adjoining pseudoprograms. The bases of two adjoining pseudoprograms differ only in one restraint vector. To obtain a new pseudoprogram, we compute the coefficients in the expansion of the constraint vector in terms of the basis vectors of the given pseudoprogram. The basis vectors corresponding to negative coefficients can be eliminated from the basis. Although in most cases there are several such basis vectors, only one is eliminated. Another question is whether it is possible to eliminate from the pseudoprogram basis, several suitable vectors simultaneously.

The present section deals with these questions. The answer to both questions is positive, and for problems with large dimensions application of the modified technique outlined in what follows has concrete advantages. The principal part of the section is devoted to the simplex method; modification of the dual simplex method is given more briefly. Some of the results of this section can also be found in /2/.

2-2. The entire discussion is with reference to the linear-programming problem of maximization of the linear form

$$L(X)=\sum_{j=1}^{n}c_jx_j \tag{2.1}$$

subject to the conditions

$$\sum_{j=1}^{n}a_{ij}x_j=b_i, \qquad i=1,\ 2,\ \ldots,\ m; \tag{2.2}$$

$$x_j\geqslant 0, \qquad j=1,\ 2,\ \ldots;\ n. \tag{2.3}$$

Let $X_0=(x_1^{(0)},\ x_2^{(0)},\ \ldots,\ x_n^{(0)})$ be a support program of problem (2.1)–(2.3) with basis $A_{s_1},\ A_{s_2},\ \ldots, A_{s_m}$. Here, as always, $A_j=(a_{1j},\ a_{2j},\ \ldots,\ a_{nj})^T$ is the j-th restraint vector of problem (2.1)–(2.3).

We rewrite system (2.2) in the equivalent form expressing the basis variables of the given program in terms of the other unknowns of the system (see Chapter 5, 4-1):

$$x_{s_i}=x_{i0}-\sum_{j\notin I_{X_0}}x_{ij}x_j, \qquad i=1,\ 2,\ \ldots,\ m. \tag{2.4}$$

Here $x_{1j},\ x_{2j},\ \ldots,\ x_{mj}$ are the coefficients in the expansion of the vector A_j in terms of the basis vectors $A_{s_1},\ A_{s_2}\ \ldots,\ A_{s_m}$ of program X_0 ($j=0,\ 1,\ \ldots,\ n;\ A_0=B$).

Let I_{X_0} be the set of indices $s_1, s_2, \ldots, s_m$ corresponding to the basis vectors of program X_0.

If we introduce the expressions for the variables $x_{s_i}, i=1, 2, \ldots, m$ into (2.1), we obtain (see Chapter 5, (4.7))

$$L(X)=L(X_0)-\sum_{j \notin I_{X_0}} \Delta_j x_j, \tag{2.5}$$

where

$$\Delta_j=\sum_{i=1}^{m} x_{ij} c_{s_i}-c_j, \qquad j=1, 2, \ldots, n.$$

If $\Delta_j \geqslant 0$ for $j \notin I_{X_0}$, program X_0 is optimal. Now let some of the Δ_j be negative. Following the simplex method, we choose one of these parameters, say $\Delta_k < 0$. The value of the extrabasis variables x_k begins to increase and the other extrabasis variables retain their zero values. The value of the basis variables is determined from formula (2.4).

Let the current value of the k-th variable be θ, then restraints (2.3) are satisfied if and only if

$$x_{i0}-\theta x_{ik} \geqslant 0, \qquad i=1, 2, \ldots, m; \ \theta \geqslant 0. \tag{2.6}$$

When conditions (2.6) are satisfied, the vector

$$X(\theta)=(x_1(\theta), \ x_2(\theta), \ \ldots, \ x_n(\theta)),$$

where

$$x_j(\theta)=\begin{cases} x_{i0}-\theta x_{ik}, & j=s_i \ (i=1, 2, \ldots, m), \\ \theta, & j=k, \\ x_j^{(0)} & \text{in other cases,} \end{cases}$$

is a feasible program of problem (2.1)–(2.3).

The transformation from program X_0 to program $X(\theta)$ increases the linear form (2.1) by

$$L(X(\theta))-L(X_0)=-\Delta_k \theta. \tag{2.7}$$

It is obviously best to choose such a value θ_0 of the variable θ so that the linear function (2.7) is maximized subject to (2.6). The value θ_0 is thus a solution of problem (2.6), (2.7) in one variable (θ).

Now let there be several negative Δ_j. We shall try to introduce several of the restraint vectors A_j $(j \notin I_{X_0}, \ \Delta_j < 0)$ simultaneously into the basis of program X_0.

Let E be the set of indices of the vectors A_μ chosen to be introduced into the basis. Let $\theta_\mu \ (\mu \in E)$ be the current value of the variable corresponding to the vector A_μ. It follows from (2.4) that for fixed θ_μ, $\mu \in E$, the values of the basis variables X_0 of x_{s_i} are equal to

$$x_{i0}-\sum_{\mu \in E} x_{i\mu}\theta_\mu.$$

The parameters θ_μ, $\mu \in E$, thus define a transformation of program X_0 into vector $X(\theta)=(x_1(\theta), \ \ldots, \ x_n(\theta))$, where

$$x_j(\theta)=\begin{cases} x_{i0}-\sum\limits_{\mu \in E} x_{i\mu}\theta_\mu, & j=s_i \ (i=1, 2, \ldots, m), \\ \theta_j, & j \in E, \\ 0 & \text{in other cases,} \end{cases} \tag{2.8}$$

Here θ indicates the set of parameters θ_μ for $\mu \in E$.

By definition, the vector $X(\theta)$ satisfies equalities (2.2) for any θ_μ, $\mu \in E_X$. Therefore, the inequalities

$$x_{i0} - \sum_{\mu \in E} x_{i\mu}\theta_\mu \geqslant 0, \qquad i = 1, 2, \ldots, m, \tag{2.9}$$

and

$$\theta_\mu \geqslant 0, \quad \mu \in E, \tag{2.10}$$

are necessary and sufficient for the vector $X(\theta)$ to be a feasible program of problem (2.1)–(2.3).

It follows from equality (2.5) that

$$L(X(\theta)) - L(X_0) = -\sum_{\mu \in E} \Delta_\mu \theta_\mu. \tag{2.11}$$

The parameters θ_μ, $\mu \in E$, should be so chosen that under transformation to the new program $X(\theta)$, linear form (2.1) acquires the maximum possible increment. Hence, the required values of the parameters θ_μ, $\mu \in E$, are determined by maximizing (2.11) whose variables are subject to conditions (2.9) and (2.10).

We shall refer to problem (2.9)–(2.11) as the auxiliary problem associated with program X_0 and set E. Let t be the number of variables in the auxiliary problem, equal to the number of elements in set E. The $m \times t$ restraint matrix $A_{X_0,E}$ of the auxiliary problem is constructed of elements $x_{i\mu}$.

2-3. Problem (2.9)–(2.11) is not given in canonical form. It can, therefore, be solved by the modified simplex method given in the preceding section. According to the general definition in 1-1, to each support program of problem (2.9)–(2.11) there corresponds a regular square submatrix of matrix $A_{X_0,E}$, the basis of the support program. The order $\bar{\tau}$ of the basis ranges from 0 to $\min(t, m)$. The number of variables t in the auxiliary problem (2.9)–(2.11) is usually taken less than m. Therefore

$$0 \leqslant \bar{\tau} \leqslant t.$$

Observe that the case $\bar{\tau} = 0$ (the set of basis elements is empty) is possible in this problem. It corresponds to the support program $\theta_\mu = 0$ for $\mu \in E$.

Taking $\theta_\mu = 0$ for $\mu \in E$ as the initial support program, we solve problem (2.9)–(2.11) by the simplex method. Since $\Delta_\mu < 0$ for $\mu \in E$, the first iteration cannot terminate in case (a). Hence, solution of the auxiliary problem is terminated either when unsolvability of the problem is established, or when an optimal support program is obtained with basis of order τ, where

$$1 \leqslant \tau \leqslant t.$$

The first possibility implies unsolvability of the initial problem (2.1)–(2.3). Consider the second possibility.

Let the basis of the support solution $\bar{\theta}_\mu$, $\mu \in E$, of the auxiliary problem comprise the elements which in matrix $A_{X_0,E}$ are the intersections of the rows and the columns with the indices $r_1, r_2, \ldots, r_\tau$ and $k_1, k_2, \ldots, k_\tau$, respectively. Let $X' = (x_1', x_2', \ldots, x_n')$ be a program of problem (2.1)–(2.3) determined from formula (2.8), where the parameters θ_μ are replaced by $\bar{\theta}_\mu$ $(\mu \in E)$.

Obviously

$$x_{s_i}' = x_{i0} - \sum_{\mu \in E} x_{i\mu}\bar{\theta}_\mu = 0 \text{ for } i = r_1, r_2, \ldots, r_\tau;$$
$$x_j' = 0, \quad \text{if} \quad j \notin I_{X_0} \quad \text{and} \quad j \neq k_1, k_2, \ldots, k_\tau.$$

Let $I_{X'}$ be the set of indices of restraint vectors obtained from I_{X_0} when the indices s_i for $i=r_1, r_2, \ldots, r_\tau$ are replaced by $k_1, k_2, \ldots, k_\tau$.

To show that X' is a support program, we must prove linear independence of the vectors A_j, $j\in I_{X'}$.

Let

$$X_j = A_{X_0}^{-1}A_j = (x_{1j}, x_{2j}, \ldots, x_{mj})^T,$$

where A_{X_0} is a matrix constructed of the basis vectors of program X_0. Take the determinant $D_{X'}$ constructed of the components of X_j, $j\in I_{X'}$. Expanding the determinant $D_{X'}$ in columns

$$X_{s_i} = (\underbrace{0, 0, \ldots, 0, 1}_{i}, 0, \ldots, 0)^T,$$

$$1\leqslant i\leqslant m, \quad i\neq r_\lambda, \quad 1\leqslant \lambda\leqslant \tau,$$

we have

$$D_{X'} = \pm D_0,$$

where D_0 is the determinant of the basis of the solution of auxiliary problem (2.9)–(2.11). However, $D_0\neq 0$. Hence $D_{X'}\neq 0$, and the vectors X_j, $j\in I_{X'}$ are linearly independent.

Further, since $A_j = A_{X_0}X_j$ and $|A_{X_0}|\neq 0$, we conclude that the system A_j for $j\in I_{X'}$ is linearly independent. We have thus established that X' is a support program.

The solution of the auxiliary problem gives

(a) the indices $j\in E$ of the vectors A_j which may be introduced into the basis $(j=k_\lambda, \lambda=1, 2, \ldots, \tau)$;

(b) the indices s_i of the vectors A_{s_i} to be eliminated from the basis $(i=r_\lambda, \lambda=1, 2, \ldots, \tau)$.

Observe that the number τ of vectors introduced into the basis may be less than the number t of elements in set E. However, we have seen that $\tau\geqslant 1$. Therefore, at least one of the basis positions will have a new vector. In most cases the number of basis positions into which new vectors are introduced is greater than one. All the operations needed for transformation from program X_0 to program X' will be called the l a r g e i t e r a t i o n. A single step of the simplex method applied for solving the auxiliary problem will be called a s m a l l i t e r a t i o n. Before the large iteration associated with the support program X' can be started, we must either know the coefficients in the expansion of the restraint vectors in terms of the basis of program X' (first algorithm) or the inverse of the basis matrix of program X' (second algorithm).

2-4. Consider the second algorithm. We shall derive recurrence formulas for the elements of the matrices

$$A_{X_0}^{-1} = \|e_{ij}\|_m, \quad A_{X'}^{-1} = \|e'_{ij}\|_m.$$

The matrices A_{X_0} and $A_{X'}$, i.e., the bases of programs X_0 and X', differ in τ vectors. To simplify notations, let us assume that the first τ positions in A_{X_0} are replaced, i.e., $r_1=1, \ldots, r_\tau=\tau$. This can always be attained by performing a suitable permutation of the columns of matrix A_{X_0} and the rows of $A_{X_0}^{-1}$. Thus,

$$A_{X'} = (A_{k_1}, A_{k_2}, \ldots, A_{k_\tau}, A_{s_{\tau+1}}, \ldots, A_{s_m}).$$

Multiplying $A_{X'}$ on the left by $A_{X_0}^{-1}$, we have

$$A_{X_0}^{-1}A_{X'} = (X_{k_1}, X_{k_2}, \ldots, X_{k_\tau}, e_{\tau+1}, \ldots, e_m), \tag{2.12}$$

where

$$X_j = A_{X_0}^{-1} A_j,\quad e_i = (\underbrace{0,\ 0,\ \ldots,\ 0,\ 1,}_{i}\ 0,\ \ldots,\ 0)^T.$$

The matrix $(X_{k_1}, X_{k_2}, \ldots, X_{k_\tau})$ is constructed of the coefficients in the expansion of the vectors $A_{k_1}, A_{k_2}, \ldots, A_{k_\tau}$, introduced into the basis in terms of the basis vectors of program X_0.

Let $\bar{B}_\tau$ be the square matrix of order τ constructed of the first τ rows of the matrix $(X_{k_1}, \ldots, X_{k_\tau})$ (the elements of $\bar{B}_\tau$ are the coefficients in the expansion of the vectors introduced into the basis which correspond to the vectors eliminated from the basis). Let $\bar{\bar{B}}_{m-\tau,\tau}$ be the $(m-\tau)\times\tau$, matrix comprising the remaining m rows of the matrix $(X_{k_1}, X_{k_2}, \ldots, X_{k_\tau})$.

With these notations equality (2.12) can be rewritten in the form

$$A_{X_0}^{-1} A_{X'} = \begin{Vmatrix} \bar{B}_\tau & O_{\tau,\, m-\tau} \\ \bar{\bar{B}}_{m-\tau,\,\tau} & E_{m-\tau} \end{Vmatrix}, \tag{2.13}$$

where $O_{\tau,\, m-\tau}$ is the $\tau\times(m-\tau)$ zero matrix, and $E_{m-\tau}$ is the unit matrix of order $m-\tau$.. By direct matrix multiplication, we can easily verify that

$$D = \begin{Vmatrix} \bar{B}_\tau & O_{\tau,\, m-\tau} \\ \bar{\bar{B}}_{m-\tau,\,\tau} & E_{m-\tau} \end{Vmatrix}^{-1} = \begin{Vmatrix} \bar{B}_l^{-1} & O_{\tau,\, m-\tau} \\ -\bar{\bar{B}}_{m-\tau,\,\tau}\bar{B}_\tau^{-1} & E_{m-\tau} \end{Vmatrix}. \tag{2.14}$$

Applying (2.13) and (2.14), we obtain

$$D A_{X_0}^{-1} A_{X'} = E_m.$$

Hence

$$A_{X'}^{-1} = D A_{X_0}^{-1}. \tag{2.15}$$

Equality (2.15) establishes the desired relationship between the matrices $A_{X'}^{-1}$ and $A_{X_0}^{-1}$.

We divide the matrix $A_{X_0}^{-1} = \|e_{ij}\|_m$ into two: (1) a matrix comprising the first τ rows of $A_{X_0}^{-1}$; (2) a matrix comprising the remaining $m-\tau$ rows. The order of these matrices are $\tau\times m$, $\|e_{ij}\|_{\tau,\, m}$, and $(m-\tau)\times m$, $\|e_{l+\tau,\, j}\|_{m-\tau,\, m}$, respectively. In the same way, we divide the matrix $A_{X'}^{-1} = \|e'_{ij}\|_m$ into the matrices $\|e'_{ij}\|_{\tau,\, m}$ and $\|e'_{l+\tau,\, j}\|_{m-\tau,\,\tau}$. The special structure of matrix D, as defined by (2.14), makes it possible to rewrite formulas (2.15). Thus:

$$\|e'_{l+\tau,\, j}\|_{m-\tau,\,\tau} = \|e_{l+\tau,\, j}\|_{m-\tau,\, m} - \bar{\bar{B}}_{m-\tau,\,\tau}\bar{B}_\tau^{-1}\|e_{ij}\|_{\tau,\, m}; \tag{2.16}$$

$$\|e'_{ij}\|_{\tau,\, m} = \bar{B}_\tau^{-1}\|e_{ij}\|_{\tau,\, m}. \tag{2.17}$$

These formulas are generalizations of recurrence formulas (1.21) in Chapter 5, which were widely used in stating the algorithms of linear-programming methods. The formulas there are a particular case of (2.16), (2.17) for $\tau = 1$.

We now give a brief outline of the sequence of operations involved in each large iteration under the second algorithm.

To begin with we have the support program X_0 with the basis $(A_{s_1}, A_{s_2}, \ldots, \ldots, A_{s_m}) = A_{X_0}$ and the inverse matrix

$$A_{X_0}^{-1} = \|e_{ij}\|_m.$$

We compute

$$\lambda_l = \sum_{\alpha=1}^{m} c_{s_\alpha} e_{\alpha l},\qquad l = 1,\ 2,\ \ldots,\ m,$$

and then

$$\Delta_j = \sum_{i=1}^{m} a_{ij}\lambda_i - c_j,\qquad j \notin I_{X_0}.$$

If all the Δ_j are nonnegative, X_0 is an optimal program. If some of the Δ_j are negative, we proceed with the construction of the new support program X'. We choose several (t) restraint vectors which may be introduced into the basis (the set of indices of these vectors is denoted by E). The expediency of introducing a vector A_j into the basis is generally determined by Δ_j the evaluation of A_j with respect to the previous basis. However, the system of vectors to be introduced into the basis can be chosen also on the basis of other considerations. The choice of the number t of the elements in set E depends, to a considerable extent, on the computational means available, and generally $t < m$ (we shall return to this problem in 2-5).

The set E specifies auxiliary problem (2.9)–(2.11) in t nonnegative variables with m inequality restraints. The parameters x_{ij} of the auxiliary problem are computed with the aid of the matrix $\|e_{ij}\|_m$:

$$x_{ij} = \sum_{\alpha=1}^{m} a_{\alpha j} e_{i\alpha} \qquad (j \in E).$$

Problem (2.9)–(2.11) is solved by the modified simplex method described in the preceding section. In the initial program of the auxiliary problem, all the variables are zero, and the basis of this program is empty. In each iteration of the auxiliary problem we compute (from recurrence formulas) the inverse of the basis matrix of the corresponding program. The solution procedure is terminated only when unsolvability of the auxiliary program is established, or its optimal program with basis $\bar{B}_\tau$ is found. The former case implies unsolvability of problem (2.1)–(2.3), and in the latter we pass to a new support program of the problem in question.

Let the basis $\bar{B}_\tau$ of the optimal program $\bar{\theta}_\mu$, $\mu \in E$ of problem (2.9)–(2.11) comprise the elements of the restraint matrix of the auxiliary problem which are the intersections of the rows and the columns with the indices $r_1, r_2, \ldots, r_\tau$ and $k_1, k_2, \ldots, k_\tau$ respectively. In this case the basis $A_{X'}$ of the new support program X' of problem (2.1)–(2.3) is obtained from A_{X_0} when the vectors $r_1, r_2, \ldots, r_\tau$ are inserted in the positions $A_{k_1}, \ldots, A_{k_\tau}$; the components of program X' are computed from (2.8), where θ_μ are replaced by $\bar{\theta}_\mu$.

Together with the optimal program of the auxiliary problem we also compute the matrix $\bar{B}_\tau^{-1}$. Therefore, the inverse

$$A_{X'}^{-1} = \|e'_{ij}\|_m$$

of the basis matrix of program X' can be determined from recurrence formulas (2.16), (2.17). As a result we obtain program X' and matrix $A_{X'}^{-1}$, both of which are essential for proceeding with the next large iteration.

2-5. We now make some comments which will help the reader to see more clearly the essence and the advantages of the modified simplex method described here. We consider first the geometrical interpretation of the large iteration, limiting the discussion to the first geometrical interpretation. The support program X_0 corresponds to a vertex of the polyhedral restraint set of problem (2.1)–(2.3) (the set M). Having established nonoptimality of program X_0, we form the set E of the indices of restraint vectors chosen to be introduced in the new basis. The set E constitutes the auxiliary problem of the given iteration. The vertex X is the intersection of hyperplane (2.2) and

$$x_j = 0 \quad \text{for } j \notin I_{X_0}. \tag{2.18}$$

Having isolated the set E, we replace restraints (2.18) for $j \in E$ by $x_j \geqslant 0$. Hence, the polyhedral restraint set M_E of the auxiliary problem coincides with the face of the set M passing through the vertex X_0. The auxiliary problem entails maximization of the linear form of the principal problem on the set M_E. The dimensionality ϱ_E of the set M_E is at most t. In the nondegenerate case $\varrho_E = t$. The point X_0 is a vertex of the set M_E. The auxiliary problem is solved by examining the adjoining vertices of the set M_E, starting with the vertex X_0. This procedure is carried out following the rules of the simplex method and therefore (in the nondegenerate case) the linear form of problem (2.1)–(2.3) increases monotonically.

The transition from a given vertex of the set M_E to another (adjoining) vertex is a small iteration. Each vertex of M_E is also a vertex of the set M. Therefore, having derived an optimal support program of the auxiliary problem, we arrive at a new support program X' of the principal problem, the result of the given large iteration.

For $t=1$ and a nondegenerate program X_0, the set M_E is an edge of the set M.

The bounded set M_E has two vertices. The auxiliary problem is, therefore, solved by a single iteration. This case applies when the unmodified form of the simplex method is employed.

The auxiliary problem is obtained from the principal problem in which some of the variables are taken to be zero. The zero variables for each auxiliary problem can be chosen by different methods.

When the simplex method is used, one is naturally guided by the evaluations Δ_j of the restraint vectors A_j with respect to the given basis taking as zero those x_j for which $\Delta_j > \Delta$, where Δ is some number. The Hungarian method and its modifications are also based on solving a sequence of auxiliary problems. Here, however, $x_j = 0$ are chosen proceeding from the current program of the dual problem.

The modified simplex method described here removes some of the restrictions imposed on the dimensions of the problem by the capacity of the operative memory of the computer used for the solution. The restraint matrix with large dimensions can be stored in the external memory of the machine.

During the large iteration it is advisable to store in the working memory of the computer only the restraints of the auxiliary problem whose dimensions are $m \times t$. This makes it possible to carry out all operations related to the external memory at the end of the large iteration. A lot of the time required by the machine for solving a problem is spent in addressing external memory devices. Therefore, reducing the number of times necessary to address these devices greatly reduces the time needed for solution.

When forming the auxiliary problem, we may vary the parameter t, the number of variables of the auxiliary problem. The dimensions of the operative memory bounds t from above. In the first stages of solving the principal problem, when the program available is still far from the optimum, the increase in the linear form in a single iteration will be larger, the larger t. Therefore, increasing t, we reduce the number of large iterations and therefore the number of addresses to the external memory.

This, however, obviously increases the number of small iterations required for solving the auxiliary problem. Increase in the dimensions of the auxiliary problem affects only slightly the time required for solving the

entire problem, since small iterations do not entail addressing the external memory. Thus, in the first stages of the solution t should be taken as large as possible. In the last stages of solving the principal problem, it is advisable to take t smaller, since then the increase in the linear-form does not depend much on t.

When problems with large dimensions are solved by finite linear-programming method, the number of iterations may be considerable. Therefore, to obtain a solution with some degree of accuracy we should

(a) either carry out all computations to a sufficient number of significant figures,

(b) or, at fixed intervals, compute the current parameters directly, without resorting to the recurrence formulas.

The applicability of (a) is limited by the dimensions of the operative memory. If (b) is carried out, the time for solution is substantially increased. The modified simplex method to a large extent reduces the effect of rounding-off errors. This is so because in each large iteration we are concerned with a problem with relatively small dimensions. When passing to the successive large iteration, it is sometimes expedient to obtain the problem parameters by computing directly the corresponding inverse matrix.

2-6. We now consider the possibility of simultaneous elimination of several vectors from the pseudoprogram basis in the dual simplex method. We give but a brief description of the modified dual simplex method. Nevertheless the reader, having thoroughly acquainted himself with the preceding articles of this section, will be able to fill in the missing details.

Let the pseudoprogram $X_0=(x_1^{(0)}, \ldots, x_n^{(0)}$ of problem (2.1)–(2.3) be given, its basis being $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$. The pseudoprogram X corresponds to some support program $Y_0=(y_1^{(0)}, y_2^{(0)}, \ldots, y_m^{(0)})$ of the dual problem of problem (2.1)–(2.3).

Let
$$A_j=\sum_{i=1}^{m} x_{ij}A_{s_i}, \qquad j=0,\ 1,\ \ldots,\ n;$$
$$\Delta_j=\sum_{i=1}^{m} a_{ij}y_i^{(0)}-c_j=\sum_{i=1}^{m} c_{s_i}x_{ij}-c_j,\ j=1,\ 2,\ldots,\ n.$$

Thus
$$x_{s_i}^{(0)}=x_{i0},\quad i=1,\ 2,\ \ldots,\ m;\quad x_j^{(0)}=0\quad \text{for}\ j\neq s_i.$$

Let some of the basis components x_{i0} of the pseudoprogram X_0 be negative. This indicates that X_0 must be modified by refining the system of preliminary evaluations of Y_0.

We take some pseudoprogram-basis vectors with negative basis components. Let these vectors occupy the positions $r_1, r_2, \ldots, r_t$ $(t<m)$ in the basis. To establish which of these vectors should be eliminated from the basis and which vectors should be introduced, we must solve the following auxiliary problem:

Maximize the linear form
$$-\sum_{j=1}^{n}\Delta_j x_j \tag{2.19}$$
subject to the conditions
$$\sum_{j=1}^{n} x_{ij}x_j=x_{i0}, \qquad i=r_1,\ r_2,\ \ldots,\ r_t; \tag{2.20}$$
$$x_j\geqslant 0, \qquad j=1,\ 2,\ \ldots,\ n. \tag{2.21}$$

Let the components of the vector $\bar{X}_0=(\bar{x}_1^{(0)}, \bar{x}_2^{(0)}, \dots, \bar{x}_n^{(0)})$ be defined as

$$\bar{x}_j^{(0)}=\begin{cases} x_j^{(0)}, & \text{if } j=s_\lambda \text{ for } \lambda=r_1, r_2, \dots, r_t, \\ 0 & \text{in other cases.}\end{cases}$$

Obviously, $\bar{X}_0$ is a pseudoprogram of problem (2.19)–(2.21) with the basis A_{s_λ}, $\lambda=r_1, r_2, \dots, r_t$ Therefore, to solve problem (2.19)–(2.20) it is advisable to apply the dual simplex method starting with the pseudoprogram $\bar{X}_0$.

Having solved problem (2.19)–(2.21), we obtain its optimal program with the basis $A_{k_1}, A_{k_2}, \dots, A_{k_t}$.

The basis of the new pseudoprogram of problem (2.1)–(2.3) is obtained from the previous basis by inserting the vectors $A_{k_1}, A_{k_2}, \dots, A_{k_t}$ in the positions $r_1, r_2, \dots, r_t$. Observe that some of the vectors A_{k_i} may coincide with the vectors A_{s_λ}, $\lambda=r_1, \dots, r_t$; the corresponding positions of the basis remain unchanged. This modified dual simplex method may be applied both under the first and under the second algorithms. Furthermore, it has the same computational advantages as the corresponding modification of the simplex method (see 2-5).

§ 3. Classification of the finite methods of linear programming

3-1. In previous chapters we studied in detail three finite linear-programming methods:* (a) the simplex method, (b) the dual simplex method, (c) the Hungarian method.

In different chapters we mentioned other finite methods and possible modifications of the three methods described in this book. It seems advisable to consider the various finite methods from one viewpoint, comparing the basic premises of each, the structure of each iteration, and the sequence of transitions between successive iterations.

The classification of the finite linear-programming methods admits of various approaches. The methods can be divided into groups depending on whether the primal or the dual problem, or both problems of the dual pair are solved. Linear-programming problems have obvious geometrical and economic interpretations. The methods available for solving linear extremum problems may also be classified according to each of these interpretations. There are also other formal principles according to which the finite methods can be analyzed from a single viewpoint.

The necessity of a unified approach to the various linear-programming problems led us to repeat, in the following, some general considerations which have been discussed in various chapters with reference to each method separately.

3-2. Finite methods of linear programming can, obviously, be distinguished according to whether the solution of the problem is obtained by examining programs of the primal problem, the dual problem, or both problems of the dual pair simultaneously.

Using this distinction as the basis of classification, the simplex method and its various modifications should form the first group, in which the optimal program is obtained by examining support programs of the primal problem. The solution procedure starts with an analysis of the initial

support program. In each iteration we transfer from one support program to another which, generally speaking, increases the value of the linear form.

The dual simplex method and its various modifications, obviously, form the second group of finite linear-programming methods in which the problem is solved by examining programs of the dual problem. The solution procedure starts with an analysis of the given support program of the dual problem. In each iteration we pass from a given support program of the dual problem to another or, equivalently, from a given pseudoprogram of the primal problem to another. The transition to a new program of the dual problem will, generally speaking, decrease its linear form. In other words, the transition to a successive pseudoprogram will decrease the linear form of the primal problem. The computational procedure of the dual simplex method is stated in terms of the primal problem.

The Hungarian method and its modifications, in our classification, form the third group of methods, in which both problems of the dual pair are dealt with. The solution procedure starts with an analysis of the given feasible program (not necessarily a support program) of the dual problem. In each iteration we pass from a given program of the dual problem to another. Simultaneously, in the augmented problem, which differs from the primal problem in its linear form and in having additional variables with unit restraint vectors, we pass from a given support program to another. In terms of the primal problem, each iteration of the Hungarian method is a transition from a given quasiprogram to a next quasiprogram with a smaller residue.

The characteristics of methods of the third group stand out most clearly in one of the modifications of the Hungarian method, namely the method of bilateral evaluations. Here the solution begins with a feasible (not necessarily a support) program of the dual problem and a support program of the primal problem. In each iteration we pass to a new feasible program of the dual problem and a new support program of the primal problem. In the method of bilateral evaluations each iteration reduces the program residue, i.e., the difference between the values of the linear forms of the dual and the primal problems on the corresponding programs.

These considerations show that the three methods described in this book can be taken as principal representatives of the three groups of finite methods reflecting essentially different approaches to the solution of linear-programming problems.

3-3. In Chapters 1, 2, and 3 we discussed two geometrical interpretations of the linear-programming problem and its dual problem. The second interpretation is more convenient for reviewing and comparing the various finite methods. When applying this interpretation we assume the problem to be given in canonical form.

In the $(m+1)$-dimensional space of points $U=(u_1, \ldots, u_{m+1})$ we consider a convex polyhedral cone K and a line Q. The cone K is spanned by the augmented restraint vectors $\bar{A}_j (j=1, 2, \ldots, n)$. The line Q parallel to the Ou_{m+1}-axis passes through the point $B=(b_1, \ldots, b_m, 0)$ defined by the constraint vector. A correspondence is established between the $(m+1)$-dimensional space of the points U and the n-dimensional space of points $X=(x_1, \ldots, x_n)$, according to which the cone K corresponds to points $X \geqslant 0$, and the line Q is the image of points with zero residue vector

$$E=B-AX$$

(here A is the restraint matrix and B the constraint vector). Thus, the intersection of the line Q and the cone K corresponds to points X satisfying the conditions

$$X \geqslant 0, \quad AX = B,$$

and is, consequently, the image of the domain of definition of the linear form.

In a solvable linear-programming problem, the intersection of the line Q and the cone K may either be a segment or a ray. For the sake of simplicity we shall assume that the intersection is a segment. The coordinate u_{m+1} of the point U specifies the value of the linear form on the corresponding vector X. The highest point of intersection M of the line Q and the cone K corresponds to the maximum of the linear form, and the lowest point m, to the minimum of the linear form in its domain of definition. Thus, geometrically, the linear-programming problem amounts to finding the highest (in maximization problems) point of intersection of the line Q and the cone K.

The hyperplanes spanned by the m linearly independent augmented restraint vectors and which intersect the segment Mm correspond to support programs of the problems. The hyperplanes spanned by the m linearly independent augmented restraint vectors which do not intersect the segment and extend above the cone K correspond to support programs of the dual problem.

It can be proved (see Exercises 8 and 9) that if the segment Mm is the image of the domain of definition of the linear form of the primal problem, the domain of definition of the linear form of the dual problem is mapped onto the ray MQ, the half of the line Q extending upward from the point M.

The segment Mm contains a finite number of points which are images of the support programs of the primal problem. Similarly, the ray MQ has a finite number of points corresponding to the support programs of the dual problem.

To surface points of the cone K, with the exception of M and m, correspond the vectors X with nonzero residue vectors. The surface of the cone can be divided into an upper and a lower part. The point U on the cone surface belongs to the upper part of the surface if a hyperplane separating the cone K and the positive semiaxis Ou_{m+1} can be drawn through the ray OU. The lower part of the surface is defined analogously.

Constructions similar to those described in Chapter 7, 3-1, make it possible to isolate a finite number of points on the upper part of the conical surface corresponding to quasiprograms of the maximization problem, whereas points of the lower part of the surface are images of quasiprograms of the minimization problem.

Proceeding from these geometrical considerations we can summarize the specific features of the various finite linear-programming methods.

The simplex method and its modifications correspond to examination of programs of the primal problem. In each iteration we obtain a new support program which, generally speaking, ensures a higher (in maximization problems) value of the linear form. Hence, the simplex method can be described by the moving from point to point along the segment Mm (images of the support programs of the primal problem) upwards to the point M.

The dual simplex method and its modifications correspond to examination of the support programs of the dual problem. Each iteration specifies

a transformation from a given support program of the dual problem to another support program which, generally speaking, decreases the linear form. The dual simplex method is described by moving from point to point along the ray MQ (images of the support programs of the dual problem) downward to point M.

Each iteration of the Hungarian method transforms a given quasiprogram into another quasiprogram with a smaller residue. Hence, the Hungarian method, when applied to the maximization problem, corresponds to moving from point to point along the upper part of the conical surface (images of the quasiprograms) which eventually lead us to the optimum.

There are other finite linear-programming methods which produce the maximum after a more complicated procedure. In particular, from the viewpoint of this geometrical classification of the methods, the method of bilateral evaluations discussed in Chapter 7, § 6 should be classified not as a modification of the Hungarian method, but rather as a combination of the simplex and the dual simplex methods. In the method of bilateral evaluations the optimum is approached along the line Q from without and from within the cone.

The simplex method thus corresponds to motion toward the optimum from within the cone, the dual simplex method, to motion from without the cone, and the Hungarian method, to motion over the surface of the cone. In all three cases the optimum is approached through points belonging to a finite set of points.

The geometrical classification thus suggestes that the three methods discussed in the book can justifiably be considered as the principal finite methods of linear programming.

3-4. The economic interpretation of the linear-programming problem given in Chapter 1, 7-2, and of the dual problem given in Chapter 3, 1-5, outline a somewhat different approach to the classification of finite methods.

We have seen that when a problem is stated in economic terms, it is better to write the problem restraints in the form of inequalities. The primal problem with the linear form

$$L=\sum_{j=1}^{n} c_j x_j \tag{3.1}$$

and the restraints

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i, \qquad i=1,\ 2,\ \ldots,\ m, \tag{3.2}$$

$$x_j \geqslant 0, \qquad j=1,\ 2,\ \ldots,\ n, \tag{3.3}$$

is interpreted as a scheduling problem for a plant (scheduling of times or utilization intensities of various production modes from several methods given in advance). The desired program must ensure maximum output of some homogeneous commodity with given resources of various production factors.

The following notations were adopted in (3.1)–(3.3):

x_j, the time during which the plant operates under the j-th production mode (the utilization intensity of the j-th production mode);

a_{ij}, the number of units of the i-th production factor consumed under the j-th production mode in unit time (with unit intensity);

b_i, the resource of the i-th production factor;

c_j, the number of units of the output commodity released in unit time under the j-th production mode.

In some cases it is more convenient to interpret c_j as the valuation (price, cost) of the output commodity released in unit time under the j-th production mode. Depending on the definition of the coefficients c_j, the linear form indicates the total volume or the total valuation (price, cost) of the output commodity.

Each production mode is characterized by the augmented restraint vector $\bar{A}_j$. The first m components of the vector $\bar{A}_j$ are specified by the input vector, and the last component coincides with the volume or the valuation of the output commodity.

The production scheduling problem, which is the basis of the economic interpretation of linear-programming techniques, can also be written in canonical form. To this end we introduce m additional hypothetical production modes with unit input vectors $(A_{m+i}=e_i)$ and zero productivity evaluations. Under the $m+i$-th production mode only the i-th production factor is consumed (unit of i-th factor in unit time) and nothing is produced. Hypothetical production modes are employed only in programs with excess of various production factors.

The production program is specified by the vector X whose components give the times (intensities) of utilization of the various production modes. The optimal program ensures maximum valuation of the output commodity with given resources (or, equivalently, maximum production volume of the plant).

Proceeding from the valuation of the output commodity, we may, obviously, evaluate (cost) the various production factors. The costing system $y_1, y_2, \ldots, y_m$ of the production factors, i. e., the cost program, is defined as the solution of the problem dual with respect to the production scheduling problem. This problem entails minimizing the linear form

$$\tilde{L}(Y)=\sum_{i=1}^{m} b_i y_i \tag{3.4}$$

subject to the conditions

$$\sum_{i=1}^{m} a_{ij} y_i \geqslant c_j, \qquad j=1, 2, \ldots, n, \tag{3.5}$$

$$y_i \geqslant 0, \qquad i=1, 2, \ldots, m. \tag{3.6}$$

Linear form (3.4) is interpreted as the overall costing of the resources. Restraints (3.5) require that the cost of the overall input of the production factors under each production mode does not exceed the valuation of the commodity produced in the same time. Restraints (3.6) are trivial: the costs cannot be negative.

The cost program specifies an optimal production program and, conversely, an optimal production program corresponds to a cost program, i. e., a costing system of the production factors. We remind the reader that the production factors are valued in units of cost of the output commodity.

Economically, the best production modes are those under which the value of the input is equal to the value of the output commodity. We shall refer to these methods as *paying* production methods. Nonpaying production methods are, obviously, deficit methods: the value of the output commodity produced under one of the nonpaying production modes is lower than the cost of the resources used. Nonpaying production modes do not utilize the

production possibilities in the most effective way possible. An optimal production program comprises paying production modes only.

The preceding economic interpretation of the dual pair makes it possible to classify the finite linear-programming methods in terms of the production-scheduling problem.

In the simplex method solution starts with some support program. Economically, this means that the solution of the problem starts with an analysis of some set of m nonnegative numbers $x_{s_1}, x_{s_2}, \ldots, x_{s_m}$ which specify utilization intensities (times) for each of the m production modes with linearly independent input vectors $A_{s_1}, \ldots, A_{s_m}$. The production modes corresponding to the basis vectors should, obviously, be called the basis production modes.

Writing the input vector A_j in terms of the basis input vectors amounts to finding the times (intensities) x_{ij} of utilization of the basis production modes for which the input of each of the production factors will equal the corresponding input under the j-th production mode in unit time. When the plant operates under the s_i-th production mode for x_{ij} units of time $(i=1, 2, \ldots, m)$ the utilization of all the basis production modes will ensure an output of $\sum_{i=1}^{m} c_{s_i} x_{ij}$ units of commodity. The output commodity is valued in $\sum_{i=1}^{m} c_{s_i} x_{ij}$ cost units. On the other hand, if the plant operates only under the j-th production mode in unit time, c_j units of commodity will be released (or, equivalently, the production value will be c_j cost units). Thus, given the same input of production factors, the volume and, consequently, the value of the output commodity was $\sum_{i=1}^{m} c_{s_i} x_{ij}$ in the former case, and c_j in the latter. The difference

$$\Delta_j = \sum_{i=1}^{m} c_{s_i} x_{ij} - c_j$$

gives an evaluation of the j-th production mode with respect to the given system of basis production modes. The parameter Δ_j shows the expediency of introducing the j-th production mode into the program, correspondingly reducing the utilization intensity of the basis production modes. If $\Delta_j < 0$, this change is advisable. If $\Delta_j \geqslant 0$, the change would reduce the production volume: the j-th production mode is less economic than the combination of the basis methods.

Analysis of the initial production program thus amounts to computation of the parameters Δ_j. The transition to other basis production modes will not increase the output volume if $\Delta_j \geqslant 0$ for all j. In this case, the program being analyzed is optimal. If some of the Δ_j are negative, the production program can be improved by substituting the k-th production mode with $\Delta_k < 0$ for one of the basis methods. The method to be eliminated from the basis is chosen such that the new support program can be realized ($x_j \geqslant 0$ for all j).

The new program is again tested for optimality. After a finite number of program improvements, we arrive at an optimal schedule for the plant.

We now give an economic interpretation of the dual simplex method. The solution of the problem starts with an analysis of a support program of the dual problem. A program of the dual problem can be interpreted as a system of preliminary valuations of the production factors. The system of preliminary valuations should satisfy restraints (3.5), (3.6). Moreover,

to a support program of the dual problem there correspond m paying (with respect to this program) production modes with linearly independent input vectors. We shall refer to the production modes as basis modes. Let the indices of the basis production modes be $s_1, s_2, \ldots, s_m$. Production programs which comprise only paying production methods (with respect to the given system of preliminary valuations) need not be realizable. The times x_{i_0} of utilization of the paying production modes should satisfy the conditions

$$b_\lambda = \sum_{i=1}^{m} a_{\lambda s_i} x_{i0}, \qquad \lambda = 1, 2, \ldots, m. \tag{3.7}$$

If all x_{i0} satisfying system (3.7) are nonnegative, the initial system of preliminary valuations proves to be the required "cost vector" and the corresponding production program is optimal. If some of the x_{i0} satisfying system (3.7) are negative, the production program corresponding to the preliminary valuation system of the production factors cannot be realized. The valuation system should, correspondingly, be improved. To this end, the r-th production mode corresponding to the negative x_{i0} is eliminated from among the basis modes.

The choice of a paying production mode as the r-th mode was one of the reasons that the production schedule cannot be realized. The preliminary valuations should be refined by making the r-th mode nonpaying, and retaining paying methods in all the other basis positions. Increasing the deficit

$$(\Delta_r = \sum_{i=1}^{m} a_{ir} y_i - c_r)$$

of the r-th mode, we should take care that the system $y_1, y_2, \ldots, y_m$ still satisfies restraints (3.5) and (3.6) of the preliminary valuations. The limiting system of preliminary valuations which we obtain in this procedure defines a new paying production mode which will be substituted for the r-th basis mode. The program comprising the new basis production modes should again be tested for realizability. The valuations are refined until the input vectors of paying (with respect to the current system of valuations) production modes define a realizable production program.

The economic interpretation of the Hungarian method is best illustrated by one of the modifications of the method, which was called in Chapter 7 the method of bilateral evaluations.

The solution procedure starts with an analysis of a system of preliminary valuations (a feasible program of the dual problem) and a production schedule (a support program of the primal problem). The given system of preliminary valuations specifies the total cost of the production factors used. The initial production schedule gives the total value of the output commodity. The given system of preliminary valuations and the initial production schedule are optimal, if the total cost of the production factors used is equal to the total value of the output commodity. Otherwise, the deficit of the schedule is determined, i. e., the excess of costs over income. The method amounts to gradual reduction of the deficits.

Applying only basis production modes corresponding to the initial support program and production modes which are paying with respect to the given system of preliminary valuations, we construct a production schedule minimizing the deficit of the preceding program. This schedule indicates the direction in which the system of preliminary valuations should be modified. If with modified valuations of the production factors the total value of the output commodity coincides with the production costs, the new production

schedule is optimal, and the corresponding system of preliminary valuations can be taken as the cost vector. Otherwise, another auxiliary problem is solved minimizing the deficit of the preceding schedule on the set of production modes which are paying with respect to the corresponding system of preliminary valuations or belong to the basis methods of the schedule in question. Successive modification of the preliminary valuations of production factors and of the system of basis production modes gradually reduces the deficit — the excess of costs over income — and elicits the production modes and the cost system to be adopted with optimal results under the conditions of the problem in question.

Other modifications of the Hungarian method can be given analogous economic interpretations.

In the simplex method and in various modifications thereof, we are given an initial production program. An analysis of this program enables us to establish its optimality or to indicate a method for its improvement. For the optimal program we then determine the system of valuations of the production factors, the cost vector.

In various modifications of the dual simplex method, the point of departure is a given system of preliminary valuations of the production factors. The attempts to construct a production schedule corresponding to the system of preliminary valuations gradually refine the said valuations. The optimal production program corresponds to the final valuation system.

Finally, in various modifications of the Hungarian method, the process of solution starts with a production schedule and a system of preliminary valuations which, generally speaking, are improperly correlated andproduce a nonpaying program, i. e., excess of total costs over the total value of the output commodity. Successive reduction of the deficits produces an optimal production schedule and a final valuation system of the production factors.

Economic interpretation of the linear-programming problem thus characterizes the principal finite methods as the best techniques for compiling a production schedule for a plant and valuing its resources.

3-5. In conclusion, we consider some formal features of the linear-programming methods which can be used as a key in classification.

Let the problem be given in canonical form.

Maximize the linear form

$$L = CX \tag{3.8}$$

subject to the conditions

$$AX = B, \tag{3.9}$$

$$X \geqslant 0. \tag{3.10}$$

Theorem 5. 2. proved in Chapter 3 makes it possible to replace the solution of the extremum problem (3.8)–(3.10) by computation of an n-dimensional vector X and m-dimensional vector Y satisfying the following system of linear equations and linear inequalities

$$AX = B, \tag{3.11}$$

$$X \geqslant 0, \tag{3.12}$$

$$YA \geqslant C, \tag{3.13}$$

$$CX = BY. \tag{3.14}$$

The last equality follows from the system of restraints

$$(YA)_j = c_j \text{ for } x_j \neq 0. \tag{3.15}$$

The four systems of equalities and inequalities (3.11)–(3.14) specify four groups of linear-programming methods. In each method, the transition between successive iterations amounts to testing the vectors X and Y for which three out of the four restraint systems are satisfied, and the fourth system is taken as the optimality test of the pair (X,Y).

Each iteration of the method, without violating the three restraint systems, ensures a monotonic (in a certain sense) approach to the set of points (X, Y) on which the fourth system is satisfied.

In the simplex method, the transition between successive iterations amounts to testing the support programs X of problem (3.8)–(3.10). Hence, restraints (3.11) and (3.12) are satisfied in each iteration. To each support program X there corresponds a unique vector Y satisfying restraints (3.15) and, consequently, equality (3.14), too. The simplex method thus amounts to testing the points (X, Y) on which restraints (3.11), (3.12), and (3.14) are satisfied, and restraint (3.13) is not satisfied. Inequalities (3.13) or, equivalently, the restraints

$$\Delta_j=\sum_{i=1}^{m} a_{ij}y_i-c_j=\sum_{i=1}^{m} c_{s_i}x_{ij}-c_j\geqslant 0, \qquad j=1,\ 2,\dots,\ n$$

specify the optimality test of the method.

Assume that in some iteration we have arrived at point (X_0, Y_0). The polyhedral restraint set of the dual problem is defined by inequalities (3.13). Consider in the space of variables of the dual problem the hyperplane

$$BY=CX_0(=BY_0).$$

It can readily be shown (this is left to the reader) that each iteration of the simplex method monotonically reduces the distance of the hyperplane $BY=CX_0$ from the polyhedral set $YA\geqslant C$. The hyperplane corresponding to an optimal program is a support hyperplane for the domain of definition of the linear form of the dual problem.

The transition between successive iterations in the dual simplex method corresponds to testing support programs of the dual problem. Hence, restraints (3.13) are satisfied in each iteration. Each support program Y of the dual problem specifies a pseudoprogram X of the primal problem, for which restraints (3.11) and (3.14) (a consequence of (3.15)) are satisfied. The dual simplex method thus involves testing the points (X, Y) which satisfy restraints (3.11), (3.13), and (3.14). Restraints (3.12) — nonnegativity of the pseudoprogram components — serve as the optimality test of the pseudoprogram. In each iteration the distance of the hyperplane

$$CX=BY_0\ (=CX_0)$$

from the polyhedral restraint set of the primal problem

$$AX=B,\ X\geqslant 0$$

is monotonically reduced. The hyperplane corresponding to the solution of problem (3.8)–(3.10) is a support hyperplane of the domain of definition of the primal problem.

In the Hungarian method, to each iteration corresponds a program Y of the dual problem and the associated quasiprogram X of the primal problem. The vectors X and Y satisfy restraints (3.12), (3.13), and (3.14). Restraint (3.11) does not hold. The method amounts to testing the quasiprograms X (or, equivalently, the programs Y of the dual problem) which

monotonically reduce the residual ε_0 of the quasiprogram

$$\varepsilon_0=\sum_{i=1}^{m}|(AX-B)_i|.$$

The computational procedure is simplified if, in addition, the residues are assumed to be nonnegative:

$$\varepsilon_i=(B-AX)_i\geqslant 0, \qquad i=1, 2,\ldots, m.$$

Restraint (3.11) (requirement of zero residues) is the optimality test of the quasiprogram.

The methods of the fourth group imply testing the points (X, Y) corresponding to programs of a pair of dual problems.

Thus, in each iteration of the method restraints (3.11), (3.12), and (3.13) are satisfied. Condition (3.14) is not observed. Each successive iteration monotonically reduces the difference $BY-CX$

One of the methods of the fourth group amounts to simultaneous improvement of programs of the primal and the dual problems (see Chapter 6, 8-3). The vectors X and Y represent support programs of the dual pair. Each iteration monotonically reduces the difference between the linear forms of the two problems.

Another example of a method of the fourth group is the method of bilateral evaluations. Each iteration of the method produces a support program X of the primal problem and a feasible program Y of the dual problem. The point (X, Y) satisfies restraints (3.11)-(3.13), and the difference $BY-CX$ monotonically decreases with each successive iteration.

In each of the preceding methods, we test support programs, pseudoprograms, or quasiprograms of the primal problem. The total number of support programs, pseudoprograms, and quasiprograms of a problem is finite. Thus, for any of the preceding methods the total number of points (X, Y) satisfying three out of the four restraints (3.11)-(3.14) is finite. Moreover, in nondegenerate problems (we have seen that any linear-programming problem can be artificially reduced to a nondegenerate problem) the transition between successive iterations is connected with monotonic approach to the solution of the problem. These two points ensure that an optimal program will be attained in a finite number of iterations.

Until now in this section we considered linear-programming problems in canonical form. The more common form of linear-programming problems, where the restraint systems comprise both equalities and inequalities and the nonnegativity requirement is imposed only on some of the variables, makes it possible to classify linear-programming methods in more numerous groups. Instead of the two systems (3.11) and (3.12) defining the range of the linear form of the primal problem, in the general case we have three systems:

$$A_1X=B_1;\quad A_2X\geqslant B_2;$$
$$x_j\geqslant 0, \qquad j=j_1, j_2, \ldots, j_t.$$

The number of various restraint systems defining the range of variables of the dual problem also increases (again three). Only restraints of type (3.15) remain unchanged. The solution of the linear-programming problem thus amounts to solving seven systems of linear equations and linear inequalities. In the new classification, each group comprises methods corresponding to testing the vectors (X, Y) satisfying six out of the seven restraint systems. The seventh restraint system is taken as the optimality test of

the vectors (X, Y). Each iteration of the method ensures a monotonic (in some sense) approach to optimal programs of the dual pair.

We should emphasize that the characteristics of the various methods given in this article can also be applied for classifying iterative methods of linear programming. However, in this case, the set of points (X, Y) tested is infinite. Iterative methods will therefore produce an approximate (with any predetermined degree of accuracy) solution of the problem.

The formal classification of the linear-programming method presented in this article also gives us an assurance of a uniform approach to all the methods used in solving nonlinear conditional-extremum problems and it can therefore be used in devising a general system of classification of mathematical-programming methods.

EXERCISES TO CHAPTER 8

1. Prove the equivalence of the definition of a support program in 1-1 and of the general definition given in Chapter 2, 4-3.

2. Prove the equivalence of the definition of a nondegenerate support program in 1-1 and of the corresponding general definition stated in Chapter 2, 4-7.

3. Prove that for any nondegenerate problem systems (1.27) and (1.28) each have one minimum.

4. Prove the recurrence formulas (1.41).

5. Applying the arguments of 1-6, 1-8, devise a simplex computational procedure for problem (1.1)–(1.4).

6. Draw a block diagram of the simplex algorithm for the case when several vectors are introduced simultaneously into the basis.

7. Draw a block diagram of the dual simplex algorithm for the case when several vectors are eliminated simultaneously from the basis.

8. If the linear form of the primal problem is bounded above and below in the domain of its definition, the polyhedral restraint set of the dual problem is unbounded. Prove.

9. Linear forms of conjugate problems bilaterally bounded on the sets of feasible programs of these problems retain a constant value in the domain of their definition. Prove.

10. Devise a computational procedure for solving problem (1.47)–(1.50) by the simplex method without modifying the original form of the problem.

11. Give an economic interpretation of the solution procedure of problem (1.47)–(1.50) by the simplex method.

12. Devise a general dual simplex technique for linear-programming problems in arbitrary form.

APPENDIX

MATHEMATICAL PRINCIPLES OF LINEAR PROGRAMMING

We give here some elements of linear algebra and the theory of convex sets in finite-dimensional spaces, which constitute the mathematical apparatus of the theory of linear programming.

In the first section we give the prerequisites relating to finite-dimensional vector spaces, matrices, and determinants. This section is intended for reference purposes, all propositions are given without proof. The corresponding details can be found, e.g., in /121/. The next section (§ 2) deals with systems of linear equations. Particular attention is given to the geometrical interpretation of the propositions. In particular, in § 2 we discuss the connection between the theory of systems of linear equations and the dual definition of a linear manifold. At the end of the section we describe one finite method for solving systems of linear equations (the Gauss-Jordan complete elimination method), which is particularly significant in linear programming.

In the last section (§ 3) we give the necessary facts from the theory of convex sets in multidimensional spaces. We prove the theorem of separating hyperplane and comment on some important corollaries of this theorem. Moreover, in § 3 we give some useful propositions related to the concept of dimensionality of a convex set and to properties of its extreme points.

All the propositions of § 2 and § 3 are proved. Exceptions are those which are left for the reader in the corresponding exercises.

§ 1. Vectors, matrices, determinants

1-1. An ordered system of n real numbers

$$X=(x_1, x_2, \ldots, x_n)$$

is called an n-dimensional vector. The numbers $x_1, x_2, \ldots, x_n$ are known as the components of the vector X. Vectors are denoted by upper-case letters, and their components by the corresponding lower-case letters. Two n-dimensional vectors

$$X=(x_1, x_2, \ldots, x_n) \text{ and } Y=(y_1, y_2, \ldots, y_n)$$

are said to be equal if their respective components are equal, i.e., if

$$x_i=y_i \quad \text{for } i=1, 2, \ldots, n.$$

The sum of the vectors X and Y is defined by the vector

$$Z=(z_1,\ z_2,\ \dots,\ z_n)$$

whose components are $z_i=x_i+y_i$, $i=1,\ 2,\ \dots,\ n$:

$$Z=X+Y=(x_1+y_1,\ x_2+y_2,\ \dots,\ x_n+y_n).$$

Vector addition is commutative

$$X+Y=Y+X$$

and associative

$$(X+Y)+Z=X+(Y+Z).$$

The difference of an ordered pair of vectors X and Y is defined by vector Z

$$Z=X-Y=(x_1-y_1,\ x_2-y_2,\ \dots,\ x_n-y_n).$$

A vector whose components are all zero is called the zero vector:

$$0=(0,\ 0,\ \dots,\ 0).$$

For vectors X and Y to be equal it is thus necessary and sufficient that

$$X-Y=0.$$

The product of a vector X by a real number (scalar) α is defined by

$$\alpha X=(\alpha x_1,\ \alpha x_2,\ \dots,\ \alpha x_n).$$

The properties of vector addition and scalar multiplication are:

$$\alpha_1(\alpha_2 X)=(\alpha_1\alpha_2)X \quad \text{(Associativity)},$$

$$\left.\begin{aligned}(\alpha_1+\alpha_2)&=\alpha_1 X+\alpha_2 X,\\ \alpha(X+Y)&=\alpha X+\alpha Y,\end{aligned}\right\}\text{(Distributivity)}$$

$$0X=\alpha 0=0.$$

The set of all n-dimensional vectors for which addition, subtraction, and scalar multiplication are defined as above constitutes a (real) n-dimensional vector space. The adjective in parenthesis real, indicates that the vector components and the scalars are real numbers. In the following this adjective will usually be omitted. This, however, should not lead to confusion, since we shall be dealing with real cases only.

For $n=3$ the vector space defined corresponds to the ordinary three-dimensional space. To each ordered triad $(x_1,\ x_2,\ x_3)$ in this space corresponds a point with the coordinates $x_1,\ x_2,\ x_3$ (or a vector directed to this point from the origin). Conversely, to each point of the three-dimensional space corresponds an ordered triad of real numbers consisting of the point coordinates. The elements of an n-dimensional vector space for $n=3$ (or $n=2$) can, thus, be interpreted either as points or as vectors directed from the origin of the coordinates. In this book the words vector and point are used synonymously for the elements of vector spaces.

1-2. To every pair of vectors

$$A=(a_1,\ a_2,\ \dots,\ a_n) \text{ and } B=(b_1,\ b_2,\ \dots,\ b_n)$$

there corresponds a real number

$$(A,\ B)=\sum_{i=1}^{n} a_i b_i,$$

called the scalar product, and such that:

1. $(A, B)=(B, A)$.
2. $(A_1+A_2, B)=(A_1, B)+(A_2, B)$.
3. $(\lambda A, B)=\lambda(A, B)$.
4. $(A, A)\geqslant 0$, the equality applying only when $A=0$.
5. $(A, B)^2\geqslant(A, A)(B, B)$. This inequality is generally attributed to Bunyakovskii.

Two vectors A and B are said to be orthogonal if

$$(A, B)=0.$$

In particular, the zero vector is orthogonal to any vector in the given space.

We define the distance between points in an n-dimensional space (specify the space metric) with the help of the scalar product.

The length, or norm, of a vector

$$A=(a_1, a_2, \ldots, a_n)$$

is defined by

$$|A|=+\sqrt{(A, A)}=+\sqrt{\sum_{i=1}^{n} a_i^2}.$$

According to property 4 of the scalar product, $|A|\geqslant 0$ for any vector A, $|A|=0$ applying only when $A=0$.

The distance between the points A and B is

$$\varrho(A, B)=|A-B|=\sqrt{\sum_{i=1}^{n}(a_i-b_i)^2}.$$

Let A, B, and C be arbitrary points in an n-dimensional space. Applying the properties of the scalar product, we prove the following inequality (the triangle inequality):

$$|A-B|\leqslant|A-C|+|C-B|.$$

The triangle inequality, like the Bunyakovskii inequality, has a clear geometrical meaning for $n=2, 3$. It is known that the scalar product of two vectors on the plane or in three-dimensional space is equal to the product of their lengths multiplied by the cosine of the angle between them. Hence, the above inequality can be rewritten in an equivalent form

$$\frac{(A, B)}{|A||B|}\leqslant 1,$$

which indicates that, in our case, the cosine of the angle is less than or equal to unity. The triangle inequality is interpreted as follows: the sum of two sides of a triangle is no less than the length of the third side.

An n-dimensional vector space in which the scalar product has been defined (and, hence, a metric has been specified) is generally called Euclidean. In the following an n-dimensional Euclidean space will be denoted by E_n.

We say that a sequence of vectors $X_1, X_2, \ldots, X_k, \ldots$ in E_n converges to the vector X in E_n ($\lim\limits_{k\to\infty} X_k=X$) if for any $\varepsilon>0$ there exists a natural number $N(\varepsilon)$ so that for all $l\geqslant N(\varepsilon)$

$$|X-X_l|<\varepsilon.$$

It follows from the definition of the norm that the sequence $\{X_k\}$ converges

to X if and and only if

$$\lim_{k\to\infty} x_s^{(k)} = x_s, \quad s=1, 2, \ldots, n.$$

Here

$$X_k = (x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}) \qquad k=1, 2, \ldots,$$
$$X = (x_1, x_2, \ldots, x_n).$$

1-3. A rectangular array of numbers

$$A = \begin{Vmatrix} a_{11} & a_{12} & \ldots & a_{1j} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2j} & \ldots & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{i1} & a_{i2} & \ldots & a_{ij} & \ldots & a_{in} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{m1} & a_{m2} & \ldots & a_{mj} & \ldots & a_{mn} \end{Vmatrix}, \tag{1.1}$$

arranged in m rows and n columns is called a $m\times n$ matrix.

The entries in matrix A are called the elements of this matrix. Each element has an ordered pair of subscripts. The first subscript indicates the row number, and the second the column number; the intersection of this row and this column is the given element. For example, the element a_{ij} is the intersection of i-th row and j-th column of matrix A.

The matrix (1.1) is generally written more concisely, thus:

$$A = \| a_{ij} \|_{m,n}. \tag{1.2}$$

If the dimension of the matrix A is given in the text, the subscripts m and n in (1.2) may be omitted. The matrix A is sometimes written as

$$A = (A_1, A_2, \ldots, A_n),$$

where A_j is the j-th column of matrix A, or

$$A = (A^{(1)}, A^{(2)}, \ldots, A^{(m)}),$$

where $A^{(i)}$ is the i-th row of matrix A.

Matrix A is called square if $m=n$ and is said to be of order n. In concise notations, the order of a square matrix is generally indicated by a subscript on the right of the double vertical bar:

$$\| a_{ij} \|_n.$$

The elements a_{ii}, $i=1, 2, \ldots, n$ of the square matrix $\| a_{ij} \|_n$ constitute the main diagonal.

If the matrix A has only one row or one column ($m=1$ or $n=1$), it may be considered as a vector. In the first case,

$$A = (a_{11}, a_{12}, \ldots, a_{1n})$$

— an n-dimensional row vector, and in the latter—

$$A = \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{pmatrix}$$

— an m-dimensional column vector.

We next define the operations of matrix addition and multiplication.

The sum of any two matrices $A=\|a_{ij}\|_{m,n}$ and $B=\|b_{ij}\|_{m,n}$ is given by the matrix

$$C=\|c_{ij}\|_{m,n}=\|a_{ij}+b_{ij}\|_{m,n}.$$

Any element of the matrix $C=A+B$ is equal to the sum of the corresponding elements of the matrices A and B. Addition is defined only for matrices with the same dimensions.

Consider the two matrices, $S=\|s_{ij}\|_{m,k}$ and $T=\|t_{ij}\|_{k,n}$. The number of columns of S is equal to the number of rows of T. The product of matrix S by matrix T is given by

$$D=ST=\|d_{ij}\|_{m,n},$$

where

$$d_{ij}=\sum_{\lambda=1}^{k}s_{i\lambda}t_{\lambda j}. \tag{1.3}$$

Multiplication is defined only on the assumption that the number of columns in S is equal to the number of rows of B. The element in the i-th row and j-th column of matrix D (the product matrix of S and T) is equal to the sum of the products of the elements of the i-th row of S multiplied by the corresponding elements of the j-th column of T.

Matrix addition and matrix multiplication have the following properties, and are often used in matrix algebra:

1. $\|a_{ij}\|_{m,n}+\|b_{ij}\|_{m,n}=\|b_{ij}\|_{m,n}+\|a_{ij}\|_{m,n}$
   (Commutativity of addition).
2. $\|a_{ij}\|_{m,n}+(\|b_{ij}\|_{m,n}+\|c_{ij}\|_{m,n})=(\|a_{ij}\|_{m,n}+\|b_{ij}\|_{m,n})+\|c_{ij}\|_{m,n}$
   (Associativity of addition).
3. $(\|a_{ij}\|_{m,k}\|b_{ij}\|_{k,r})\|c_{ij}\|_{r,n}=\|a_{ij}\|_{m,k}(\|b_{ij}\|_{k,r}\|c_{ij}\|_{r,n})$
   (Associativity of multiplication).
4. $\|c_{ij}\|_{m,k}(\|a_{ij}\|_{k,n}+\|b_{ij}\|_{k,n})=\|c_{ij}\|_{m,k}\|a_{ij}\|_{k,n}+\|c_{ij}\|_{m,k}\|b_{ij}\|_{k,n};$
   $(\|a_{ij}\|_{m,k}+\|b_{ij}\|_{m,k})\|c_{ij}\|_{k,n}=\|a_{ij}\|_{m,k}\|c_{ij}\|_{k,n}+\|b_{ij}\|_{m,k}\|c_{ij}\|_{k,n}$
   (Distributivity of addition and multiplication).

It is easily verified that matrix multiplication is not commutative, i.e.,

$$\|a_{ij}\|_{k,r}\|b_{ij}\|_{k,r}\neq\|b_{ij}\|_{r,k}\|a_{ij}\|_{k,r}.$$

We shall usually deal with square matrices. We note that square matrices can be added and multiplied only if they are of the same order. Among square matrices of order n we have the unit (identity) matrix

$$I_n=\begin{Vmatrix}1 & 0 & 0 & \dots & 0\\ 0 & 1 & 0 & \dots & 0\\ 0 & 0 & 1 & \dots & 0\\ \cdot & \cdot & \cdot & \cdot & \cdot\\ 0 & 0 & 0 & \dots & 1\end{Vmatrix},$$

whose main diagonal elements are 1, and whose off-diagonal elements are all equal to zero. For any matrix A of order n

$$AI_n=I_nA=A.$$

Like vectors, matrices may be multiplied by scalars. The product

of a matrix $A=\|a_{ij}\|_{m,n}$ by a scalar α is given by

$$B=\alpha A=\|\alpha a_{ij}\|_{m,n}.$$

Thus, multiplication of a matrix by some number implies that each element of the matrix is multiplied by this number. We bring another operation carried out on a single matrix.

The transposed matrix with respect to matrix (1.1) (or the transpose of matrix (1.1)) is given by the matrix

$$A^T=\begin{Vmatrix} a_{11} & a_{21} & \dots & a_{i1} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{i2} & \dots & a_{m2} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{1j} & a_{2j} & \dots & a_{ij} & \dots & a_{mj} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{1n} & a_{2n} & \dots & a_{in} & \dots & a_{mn} \end{Vmatrix},$$

whose columns are the corresponding rows of the initial matrix A. Transposition of a matrix A is denoted by the superscript T. In particular, the column vector

$$A=\begin{pmatrix} a_{11} \\ a_{12} \\ \vdots \\ a_{1m} \end{pmatrix}$$

may be written as

$$A=(a_{11}, a_{12}, \dots, a_{1m})^T.$$

Transposition has the following properties:

$$1.\ (A+B)^T=A^T+B^T.$$
$$2.\ \ (AB)^T=B^TA^T,$$

i.e., the transpose of a sum is equal to the sum of transposed summands; the transpose of a product is equal to the product of transposed factors taken in reverse order.

1-4. Any set of n numbers $1, 2, \dots, n$ arranged in some definite order is called a permutation of n numbers (symbols). The number of permutations of n numbers is equal to $n!$

We generally say that the numbers i and j constitute an inversion in the given permutation if $i>j$ and still i precedes j in this permutation.

A permutation is said to be even, if its symbols constitute an even number of inversions, and odd otherwise. The number of inversions in a given permutation (and consequently its parity) can be determined in the following way.

Consider a permutation $I=(i_1, i_2, \dots, i_n)$. Let k_s be the number of symbols smaller than i_s which are right of the symbol i_s. The number of inversions of permutation I is equal to

$$k_1+k_2+\dots+k_{n-1}.$$

For example, the number of inversions in the permutation (3, 1, 4, 2, 6, 5) of six symbols is equal to

$$2+0+1+0+1=4.$$

This is therefore an even permutation.

These preliminary facts form an introduction to the concept of the determinant of an arbitrary square matrix.

Consider the square matrix

$$A=\left\|\begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{matrix}\right\| = \|a_{ij}\|_n. \tag{1.4}$$

Let us now write out all possible products of n elements of the matrix A in each of which there appears one and only one element from each row and each column. Each of these products can be written in the form

$$a_{1j_1}\, a_{2j_2} \dots a_{nj_n}. \tag{1.5}$$

By definition, the subscripts $j_1, j_2, \dots, j_n$ are all different and consequently constitute a permutation

$$(j_1, j_2, \dots, j_n). \tag{1.6}$$

The determinant of matrix A is obtained as the sum of all possible products (1.5). Each product has the sign $(-1)^d$ where d is 0 or 1 depending on whether the permutation (1.6) is even or odd. Each determinant has $n!$ such products.

The determinant of matrix A is generally denoted by $|A|$. Other notations are sometimes used, corresponding to the various written forms of the matrix A:

$$|A|=\left|\begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{matrix}\right| = |\,\|a_{ij}\|_n| = |(A_1, A_2, \dots, A_n)| = |(A^{(1)}, A^{(2)}, \dots, A^{(n)})|.$$

The elements, the rows, and the columns of matrix A are generally also the elements, the rows, and the columns of the determinant $|A|$.

The following properties of determinants are often used in computations.

1. $|A|=|A^T|$, i.e., the determinant of a matrix is not affected by transposition.

2. If every element of a row (or of a column) of a determinant is zero, the value of the determinant is zero.

3. If two rows (two columns) of a determinant are interchanged, the sign is changed but not the value of the determinant.

4. A determinant containing two identical rows (two identical columns) has a value of zero.

5. If every element of a row (column) of a determinant is multiplied by a number α, the value of the determinant is multiplied by α.

6. $|(A_1 A_2, \dots, A'_j + A''_j, \dots, A_n)| = |(A_1, A_2, \dots, A'_j, \dots, A_n)| + |(A_1, A_2, \dots, A''_j, \dots, A_n)|$;

$|(A^{(1)}, A^{(2)}, \dots, A^{(i)}_1 + A^{(i)}_2, \dots, A^{(n)})| = |(A^{(1)}, A^{(2)}, \dots, A^{(i)}_1, \dots, A^{(n)})| + |(A^{(1)}, A^{(2)}, \dots, A^{(i)}_2, \dots, A^{(n)}|$.

The first relationship can be stated in words as follows. If the j-th column A_j of a matrix A is equal to the sum of two vectors A'_j and A''_j, the determinant $|A|$ is equal to the sum of the determinants of matrices A' and A'' in which all the columns, except for the j-th, are the same as in matrix A, and the j-th column of A' (A'') is $A'_j (A''_j)$. The second relationship is interpreted analogously, writing rows instead of columns.

7. The value of a determinant is not changed if to one of the rows (columns) we add another row (column) multiplied by an arbitrary number.

8. Let a matrix A of the form (1.4) be given. In A cross out the i-th row and the j-th column, whose intersection is the element a_{ij}. The determinant of the new matrix obtained, of order $n-1$, is denoted by M_{ij}.

The cofactor of the element a_{ij} of matrix A is $(-1)^{i+j}M_{ij}=A_{ij}$.

The following equalities apply:

$$|A|=\sum_{j=1}^{n} a_{ij}A_{ij};$$

$$|A|=\sum_{i=1}^{n} a_{ij}A_{ij}.$$

In the first case we have an expansion of $|A|$ in the i-th row; in the second case, the expansion in the j-th column.

9. Let A and B be square matrices of the same order.

In this case

$$|AB|=|A||B|,$$

i. e., the determinant of a product of two square matrices is equal to the product of the determinants of these matrices.

1-5. A square matrix A is called regular if its determinant, $|A|$, does not equal zero. If $|A|=0$ the matrix A is singular.

Let A be a regular matrix of form (1.4). Consider the matrix $\|\tilde{a}_{ij}\|_n$ whose elements $\tilde{a}_{ij}$ are given by

$$\tilde{a}_{ij}=\frac{A_{ji}}{|A|}, \qquad i,\ j=1,\ 2,\ldots,n. \tag{1.7}$$

Here A_{ji} is the cofactor of the element a_{ji} of matrix A. Applying the fourth and the eighth properties of determinants, and also the definition of matrix multiplication, we obtain

$$A\|\tilde{a}_{ij}\|_n=\|\tilde{a}_{ij}\|_n A=I_n, \tag{1.8}$$

where I_n is the unit matrix of order n. We leave it to the reader to verify equality (1.8) (see Exercise 2).

It can be shown (see Exercise 3) that the matrix $\|\tilde{a}_{ij}\|_n$ is the only matrix satisfying (1.8).

A matrix A^{-1} is called the inverse (or the reciprocal matrix) of (or with respect to) matrix A if

$$AA^{-1}=A^{-1}A=I_n.$$

Thus, for a regular matrix A

$$A^{-1}=\|\tilde{a}_{ij}\|_n,$$

where the elements $\tilde{a}_{ij}$ are given by (1.7).

If the matrix A is singular it has no inverse. The proof of this is left to the reader (see Exercise 4).

1-6. Let $A_1,\ A_2,\ldots,A_s$ be an arbitrary set of n-dimensional vectors.

The vector

$$\alpha_1A_1+\alpha_2A_2+\ldots+\alpha_sA_s,$$

where α_i is a scalar ($i=1,\ 2,\ldots,s$) is called a linear combination of vectors $A_1,\ A_2,\ldots,A_s$ with the coefficients $\alpha_1,\ \alpha_2,\ldots,\alpha_s$. The system $A_1, A_2,\ldots,A_s$ is said to be linearly dependent if the zero vector

$$0=\underbrace{(0,\ 0,\ldots,0)}_{n}$$

2154

can be represented as a linear combination of the vectors $A_1, A_2, \ldots, A_s$ in which some of the coefficients are nonzeroes. Otherwise the system $A_1, A_2, \ldots, A_s$ is linearly independent. In other words, a system $A_1, A_2, \ldots, A_s$ is said to be linearly independent if

$$\sum_{i=1}^{s} \alpha_i A_i = 0$$

implies that $\alpha_1 = \alpha_2 = \ldots = \alpha_s = 0$. Vectors constituting a linearly independent system are said to be linearly independent.

It can easily be verified that the representation of a given vector as a linear combination of linearly independent vectors is unique.

We now give a criterion for establishing linear independence of a system of vectors; this criterion is widely used in various branches of mathematics.

For a system of vectors $A_i = (a_{i1}, a_{i2}, \ldots, a_{in})$, $i = 1, 2, \ldots, s$ to be linearly independent it is sufficient and necessary that there be a square matrix of order s,

$$\begin{Vmatrix} a_{1j_1} & a_{1j_2} & \ldots & a_{1j_s} \\ a_{2j_1} & a_{2j_2} & \ldots & a_{2j_s} \\ \ldots & \ldots & \ldots & \ldots \\ a_{sj_1} & a_{sj_2} & \ldots & a_{sj_s} \end{Vmatrix},$$

whose elements are the components of vectors A_i, such that its determinant does not vanish.

If, in particular, $s = n$, for the given system to be linearly independent it is necessary and sufficient that

$$|(A_1, A_2, \ldots, A_n)| \neq 0.$$

It follows that in an n-dimensional space there cannot exist a linearly independent system of more than n vectors. The unit vectors $e_1 = (1, 0, 0, \ldots, 0)$, $e_2 = (0, 1, 0, \ldots, 0), \ldots, e_n = (0, 0, 0, \ldots, 1)$ constitute a linearly independent system.

Analogously to a plane and a three-dimensional space, the dimensionality of a vector space is defined as the maximum number of linearly independent vectors in this space. Thus, the dimension of an n-dimensional vector space is n.

Let us now consider an arbitrary rectangular matrix A of the form (1.1).

A matrix comprising elements of the intersection of some rows and columns of A is called a submatrix of matrix A. The determinant of any square submatrix of the matrix A is called the minor of this matrix. The order of a minor is determined by the order of the corresponding submatrix.

The largest square array in A whose determinant does not vanish is called the rank of this matrix.

It follows from the criterion of linear independence of a vector system that the maximum number of linearly independent rows of a matrix A is equal to the maximum number of linearly independent columns of this matrix, both coinciding with the rank of A. The proof of this is left to the reader (see Exercise 6).

The rank of an arbitrary system of vectors is defined as the maximum number of linearly independent vectors in this system. The set of all linearly independent vectors of a system, whose number is equal to the rank of the system, forms the basis of the system. A set of linearly independent vectors of a given system constitutes a basis if and only if any

vector of the system can be represented as a linear combination of vectors of the set being considered. Necessity is obvious. Sufficiency can be proved proceeding from the criterion of linear independence of a system of vectors.

1-7. In the analysis of systems of linear equations, which is the subject of the next section, we shall use properties of some important sets of points in an n-dimensional space.

A nonempty set of n-dimensional vectors will be called a subspace of an n-dimensional vector space if the sum of any two vectors of the set and the product of any vector of the set by a scalar belong to this set.

All subspaces, obviously, contain the zero vector, since $O \cdot X = 0$ for any element X of the subspace.

A subspace is said to be r-dimensional if the maximum number of linearly independent vectors in it is equal to r. Let $A_1, A_2, \ldots, A_r$ be a system of linearly independent vectors in an n-dimensional vector space. Consider the set of all possible linear combinations of this system. It can easily be shown that it constitutes an r-dimensional subspace. On the other hand, in any r-dimensional subspace there exist r linearly independent vectors whose linear combinations span the subspace. The proof of these propositions is left to the reader (Exercise 7).

Thus, an r-dimensional subspace can be defined as the set of all linear combinations of some linearly independent system of r vectors. If $r=0$, the subspace consists of the zero vector only. If $r=n$, it coincides with the entire space. If $0<r<n$ we have intermediate subspaces containing an infinity of elements and not coinciding with the entire space.

The set of elements of an n-dimensional vector space which can be written in the form

$$X+X',$$

where X is a fixed vector, and X' belongs to some r-dimensional subspace, is called an r-dimensional linear manifold ($r \leqslant n$).

Thus, an r-dimensional linear manifold may be considered as some translation of an r-dimensional subspace. In particular, if the translation (the vector X) is the zero vector, the manifold reduces to the subspace.

A one-dimensional linear manifold will be called a line, and an $(n-1)$-dimensional linear manifold, a hyperplane.

Remembering the definition of an r-dimensional subspace, we ascertain that an r-dimensional linear manifold is actually a set of points of the form

$$X+\sum_{i=1}^{r} \alpha_i X_i,$$

where X is a fixed point in the n-dimensional space; $X_1, X_2, \ldots, X_r$ is some linearly independent system of n-dimensional vectors, and $\alpha_1, \alpha_2, \ldots, \alpha_r$ are arbitrary numbers.

§ 2. Systems of linear equations

2-1. Consider the system of linear algebraic equations

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \qquad i=1, 2, \ldots, m, \tag{2.1}$$

comprising m equations in n unknowns x_j ($j=1,2,\ldots,n$).

The matrix $A=\|a_{ij}\|_{m,n}$ whose elements are the coefficients of system (2.1) is called the matrix of coefficients of system (2.1). If we take $A_j=(a_{1j}, a_{2j}, \ldots, a_{mj})^T$ the matrix of coefficients of (2.1) can be written in the form

$$A=(A_1, A_2, \ldots, A_n).$$
$$B=(b_1, b_2, \ldots, b_n)^T.$$

The vector B is called the free-term column of the system.

System (2.1) may be rewritten in the following equivalent forms:

$$\sum_{j=1}^{n} A_j x_j=B; \tag{2.2}$$

$$AX=B, \tag{2.3}$$

where $X=(x_1, x_2, \ldots, x_n)^T$.

The matrix $\bar{A}=(A_1, A_2, \ldots, A_n, B)$ is called the augmented matrix of system (2.1). The vector $X'=(x_1', x_2', \ldots, x_n')$ is called a solution of system (2.1) if substitution of x_j for x_j' $(j=1, 2, \ldots, n)$ in the system reduces all the equations of (2.1) to identities.

A system of linear equations which has at least one solution is said to be consistent.

If the system has no solutions, it is said to be inconsistent, or contradictory. A consistent system is sometimes called solvable, and an inconsistent one, unsolvable.

In the following we give conditions for consistency of systems of linear equations and describe the characteristics of sets of solutions of consistent systems. Moreover, we present a method for solution of linear equations which is widely used in linear programming.

2-2. We shall start our discussion with the case in which the number m of equations in system (2.1) is equal to the number n of unknowns. The matrix of coefficients A is then a square matrix of order n.

Let us assume that the determinant of matrix A does not vanish. Let the vector $X'=(x_1', x_2', \ldots, x_n')^T$ be some solution of system (2.1). Using (2.3), we obtain

$$AX'=B.$$

By multiplying both parts of this equation from the left by the matrix A^{-1}, which is the inverse of the matrix A of the system (the existence of A^{-1} follows from the condition $|A|\neq 0$), we obtain the following chain of equations:

$$X'=IX'=(A^{-1}A)X'=A^{-1}B. \tag{2.4}$$

Thus, any solution of system (2.1) can be represented by (2.4). If the system (2.1) with square matrix A is consistent, it has a unique solution.

We shall show that the vector $X'=A^{-1}B$ solves the system (2.1). Indeed,

$$AX'=A(A^{-1}B)=(AA^{-1})B=IB=B.$$

The system (2.1) with $n=m$ and $|A|\neq 0$ is, thus, consistent and has a unique solution determined by (2.4).

Let us now transform (2.4). Applying (1.7), which defines the elements $\tilde{a}_{ij}$ of the inverse matrix A^{-1}, and the rule of matrix multiplication, we obtain

$$x_j'=\sum_{s=1}^{n} \tilde{a}_{js} b_s=\sum_{s=1}^{n} \frac{A_{sj}}{|A|} b_s. \tag{2.5}$$

According to the eighth property of determinants, we have

$$\sum_{s=1}^{n} A_{sj} b_s=|(A_1, A_2, \ldots, A_{j-1}, B, A_{j+1}, \ldots, A_n)|. \tag{2.6}$$

From (2.5) and (2.6) we have

$$x'_j=\frac{|(A_1, A_2, \dots, A_{j-1}, B, A_{j+1}, \dots, A_n)|}{|(A_1, A_2, \dots, A_n)|}, \ j=1,2,\dots,n, \tag{2.7}$$

which is known as Cramer's rule, and can be stated as follows:

If the square matrix A of a system of linear equations is regular, the j-th component of the solution of the system is equal to the determinant of a matrix which is obtained from A when the j-th column is replaced by the free-term column of the system divided by the determinant of matrix A.

2-3. We now deal with the general case in which no restrictions are imposed on the matrix of coefficients of the system of linear equations (2.1).

Let the rank of matrix A be r. We disregard the trivial case $r=0$, i.e., the matrix A consists of zeroes only. The basis of the system $A_1, A_2, \dots, A_n$ of vectors constituting the matrix A contains r vectors. To be specific let us assume that the vectors $A_1, A_2, \dots, A_r$ form the basis. Further, assume that the system (2.1) is consistent and that $X=(x_1, x_2, \dots, x_n)$ is one of its solutions. In this case the free-term vector B is a linear combination of the vectors A_j, $j=1, 2, \dots, n$:

$$B=\sum_{j=1}^{n} x_j A_j.$$

Since each of the vectors A_j, $j\geqslant r+1$, can be expressed in terms of the basis vector, we have

$$B=\sum_{j=1}^{r} \alpha_j A_j.$$

Hence, the vectors $A_1, A_2, \dots, A_r$ constitute a basis of system $A_1, A_2, \dots, A_n, B$ and, consequently, the rank of the augmented matrix $\tilde{A}=(A_1, A_2, \dots, A_n, B)$ of system (2.1) is also r (the rank of matrix A).

Now let the ranks of matrices A and $\tilde{A}$ coincide at the outset, both being equal to r.

If $A_1, A_2, \dots, A_r$ is a basis of the system $A_1, A_2, \dots, A_n$, there exist numbers α_i $(i=1, 2, \dots, r)$ such that

$$B=\sum_{i=1}^{r} \alpha_i A_i. \tag{2.8}$$

Otherwise the system of vectors $A_1, A_2, \dots, A_r, B$ would be linearly independent and the rank of matrix $\tilde{A}$ would be greater than r.

Equality (2.8) indicates that the vector $(\alpha_1, \alpha_2, \dots, \alpha_r, 0, \dots, 0)$ solves the system (2.1).

We have thus obtained the following criterion for the solvability of a system of linear equations.

Theorem 2.1. *For a system of linear equations to be consistent, it is necessary and sufficient that the rank of the coefficient matrix equal the rank of the augmented matrix.*

It is noteworthy that Theorem 2.1 applies also for $r=0$, since in this case the system is solvable if and only if $B=0$. Hence, the matrix $\tilde{A}$ and the matrix A both have rank zero.

Having established the criterion for solvability of a system of linear equation, we now must deal with the problem of the complete set of its solutions.

The system of equations

$$\sum_{j=1}^{n} a_{ij}x_j=0, \qquad j=1,\ 2,\ \ldots,\ m, \tag{2.9}$$

or, equivalently,

$$AX=0 \tag{2.10}$$

is called homogeneous, i.e., the free-term vector is zero. The following forms of system (2.9) are convenient for applications

$$\sum_{j=1}^{n} A_j x_j=0, \tag{2.11}$$

$$(A^{(i)},\ X)=0, \qquad i=1,\ 2,\ \ldots,\ m. \tag{2.12}$$

Here $A^{(i)}=(a_{i1},\ a_{i2},\ \ldots,\ a_{in})$ is the i-th row of matrix A.

Since the matrix of coefficients of (2.1) and of (2.9) are identical, the system (2.9) is said to be the homogeneous system corresponding to system (2.1).

Let X_1 and X_2 be two solutions of system (2.1) and $Y=X_1-X_2$. Obviously,

$$AY=AX_1-AX_2=0.$$

Hence, the vector Y is a solution of the homogeneous system (2.9) (or (2.10)). On the other hand, if Y is a solution of system (2.9), and X_1 is a solution of system (2.1), then $X_2=X_1+Y$ is also a solution of system (2.1), since

$$AX_2=AX_1+AY=B+0=B.$$

Hence, the set of all solutions (the general solution) of system (2.1) can always be represented as the sum of a particular solution of this system and the general solution of the corresponding homogeneous system (2.9).

2-4. We shall now investigate the general solution of the homogeneous system (2.9).

Consistency of system (2.9) is obvious, since the zero vector $0=(0,\ 0,\ \ldots,\ 0)$ is always a solution. Let the rank of the matrix of coefficients in (2.9) be r. This is equivalent to saying that the maximum order of a nonvanishing minor of matrix A is r. To be specific, let $|\|a_{ij}\|_r|\neq 0$. In this case the first r rows of matrix A are linearly independent.

We shall say that two systems of linear equations in n unknowns are equivalent if the general solutions of the two systems coincide.

Consider the system of equations

$$\sum_{j=1}^{n} a_{ij}x_j=0, \qquad i=1,\ 2,\ \ldots,\ r, \tag{2.13}$$

consisting of the first r equations of system (2.9). Using (2.12), we rewrite (2.13) in the form

$$(A^{(i)},\ X)=0, \qquad i=1,\ 2,\ \ldots,\ r. \tag{2.14}$$

We have

$$A^{(s)}=\sum_{i=1}^{r} \alpha_{is}A^{(i)}, \qquad s=r+1,\ \ldots,\ m.$$

Hence, if the vector X satisfies system (2.14), then

$$(A^{(s)},\ X)=\sum_{i=1}^{r} \alpha_{is}(A^{(i)},\ X)=0, \qquad s=r+1,\ \ldots,\ m.$$

We observe that systems (2.9) and (2.13) are equivalent, i.e., the general solution of one is also the general solution of the other.

We rewrite system (2.13) in the form:

$$\sum_{j=1}^{r} a_{ij}x_j = -\sum_{j=r+1}^{n} a_{ij}x_j, \qquad i=1, 2, \ldots, r. \tag{2.15}$$

By assumption, the determinant of the matrix $\|a_{ij}\|_r$ does not vanish. Hence, for any values of the unknowns x_j, $j=r+1, \ldots, n$ the first r unknowns of system (2.13) are uniquely determined; moreover, they can be calculated by Cramer's rule.

Let us introduce a system of n-dimensional vectors

$$X_s = (x_1^{(s)}, x_2^{(s)}, \ldots, \underbrace{x_r^{(s)}, 0, \ldots, 0, 1}_{s}, 0, \ldots, 0),$$

where the numbers $x_1^{(s)}, x_2^{(s)}, \ldots, x_r^{(s)}$ are determined from the system

$$\sum_{j=1}^{r} a_{ij}x_j^{(s)} = -a_{i,\, r+s}, \qquad i=1, 2, \ldots, r, \tag{2.16}$$

and the superscript s takes on the values from 1 to $n-r$.

Since the solution of system (2.15) is unique for any fixed value of its right-hand side, we obtain the following expression for an arbitrary solution $X'=(x_1', x_2', \ldots, x_n')$ of system (2.13):

$$X' = \sum_{s=1}^{n-r} x_{r+s}' X_s. \tag{2.17}$$

On the other hand, since (2.13) is a linear and homogeneous system, the vector $\sum_{s=1}^{n-r} \alpha_s X_s$ solves this system for any $\alpha_1, \alpha_2, \ldots, \alpha_{n-r}$.

The general solution of system (2.13) (and consequently also of the equivalent system (2.9)) is the set of all possible linear combinations of the vectors $X_1, X_2, \ldots, X_{n-r}$. The last $n-r$ components of the vectors X_s, $s=1, 2, \ldots, n-r$ constitute a square matrix with a nonvanishing determinant. The vectors $X_1, X_2, \ldots, X_{n-r}$ are, thus, linearly independent.

We have proved the following proposition:

Theorem 2.2. The set of all solutions (the general solution) of a homogeneous system of linear equations in n unknowns whose matrix of coefficients is of rank r constitutes an $(n-r)$-dimensional subspace (the solution subspace).

2-5. Let the system (2.1) be consistent, its matrix be of rank r, and X a solution of the system. Applying the previous relationship between the general solutions of systems (2.1) and (2.9) we ascertain that linearly independent vectors X_s, $s=1, 2, \ldots, n-r$ (these are solutions of system (2.9)) exist such that the set of all vectors

$$X + \sum_{s=1}^{n-r} \alpha_s X_s$$

for various values of the coefficients $\alpha_1, \alpha_2, \ldots, \alpha_{n-r}$ coincides with the general solution of system (2.1).

The assertion with regards to the general solution of system (2.1) can be summarized in the following theorem:

Theorem 2.3. The general solution of a consistent system of linear equation in n unknowns having a matrix of coefficients of rank r is an $(n-r)$-dimensional manifold. This manifold is obtained by translating the solution subspace of the corresponding homogeneous system by any vector satisfying the initial system.

The uniqueness of the solution of a system of linear equation follows as a corollary from Theorem 2.3.

Theorem 2.4. A solvable system of linear equations has a unique solution if and only if the rank of the matrix of coefficients is equal to the number of unknowns.

In particular, if the number of equations and the number of unknowns are equal, a necessary and sufficient condition for uniqueness of solution is that the matrix of coefficients be regular.

Consider an arbitrary ϱ-dimensional linear manifold comprising the points

$$X_0+\sum_{i=1}^{\varrho}\alpha_i X_i, \tag{2.18}$$

where $X_1, X_2, \ldots, X_\rho$ is a linearly independent system of vectors $(\varrho<n)$. We shall use the vectors $X_1, X_2, \ldots X_\rho$ to construct a homogeneous system of equations

$$(X_i, Y)=0, \qquad i=1, 2, \ldots, \varrho. \tag{2.19}$$

The components of the vector $Y=(y_1, y_2, \ldots, y_n)$ are the unknowns of the system (2.19). The rank of the matrix of coefficients of system (2.19) is ϱ. According to Theorem 2.2 the general solution of the homogeneous system (2.19) can be written as

$$\alpha_1 Y_1+\alpha Y_2+\ldots+\alpha_{n-\rho}Y_{n-\rho},$$

where Y_i, $i=1, 2, \ldots, n-\varrho$ are linearly independent vectors, and $\alpha_1, \alpha_2, \ldots, \alpha_{n-\rho}$ are arbitrary numbers. In particular,

$$(X_i, Y_j)=0$$

for $i=1, 2, \ldots, \varrho$; $j=1, 2, \ldots, n-\varrho$. Hence, the homogeneous system of equations

$$(Y_i, X)=0, \qquad i=1, 2, \ldots, n-\varrho, \tag{2.20}$$

is solved by the vectors

$$X_1, X_2, \ldots, X_\rho.$$

Since the vectors $Y_1, Y_2, \ldots, Y_{n-\rho}$ are linearly independent, we conclude that system (2.20) is of rank $n-\varrho$. Therefore, according to Theorem 2.2, the general solution of this system is a ϱ-dimensional subspace. The vectors $X_1, X_2, \ldots, X_\rho$ are linearly independent and are contained in this subspace. Hence the general solution of system (2.20) has the form

$$\alpha_1 X_1+\alpha_2 X_2+\ldots+\alpha_\rho X_\rho. \tag{2.21}$$

We now introduce the following system of linear equations:

$$(Y_i, X)=(Y_i, X_0), \qquad i=1, 2, \ldots, n-\varrho. \tag{2.22}$$

Obviously, X_0 is one of the solutions of system (2.22). Theorem 2.3 and (2.21) which represent the general solution of system (2.20) show that the expression (2.18) is the general solution of system (2.22).

We have thus established the converse of Theorem 2.3.

Theorem 2.5. An arbitrary ϱ-dimensional linear manifold represents the general solution of some system of linear equations in n unknowns with matrix of coefficients of rank $n-\varrho$.

Consider one linear equation in n unknowns:

$$\sum_{j=1}^{n} a_j x_j = b. \tag{2.23}$$

Let the vector $A=(a_1, a_2, \ldots, a_n)$ be any but the zero vector. It follows from Theorem 2.3 that the general solution of equation (2.23) is a $(n-1)$-dimensional linear manifold, i. e., a hyperplane.

On the other hand, according to Theorem 2.5 any hyperplane is a general solution of some equation of the form (2.23) with vector $A=(a_1, a_2, \ldots, a_n)\neq 0$. From these considerations we obtain another definition of a hyperplane.

A hyperplane is a set of points in an n-dimensional space satisfying a linear equation of the form (2.23), where $A=(a_1, a_2, \ldots, a_n)\neq 0$.

Equation (2.23) is the equation of a hyperplane. The vector $A\neq 0$ is called the direction vector of the hyperplane specified by (2.23).

Let X_0 be an arbitrary point on the hyperplane (2.23) (the hyperplane defined by (2.23)). Hyperplane (2.23) may, obviously, be considered as the set of all points which can be represented in the form X_0+X', where $(A, X')=0$. The hyperplane (2.23) is, thus, a translation, by X_0, of the set of vectors X' orthogonal to the direction vector A. Therefore we can say that a hyperplane is orthogonal to its direction vector.

In discussing linear equations, we may always assume that not all the coefficients of each equation are zero. Indeed, if the system contains an equation

$$0X_1+0X_2+\ldots+0X_n=b,$$

then for $b\neq 0$ the system is inconsistent, and for $b=0$ this equation does not impose any restraints on the vector X and can, therefore, be omitted.

Hence, the general solution of a consistent system of linear equations is the intersection (the common area) of several hyperplanes (depending on the number of equations in the system).

We shall say that hyperplanes are linearly independent if their direction vectors constitute a linearly independent system.

Using Theorems 2.3 and 2.5 we can define a linear manifold in a different manner.

The intersection of r linearly independent hyperplanes (provided it is nonempty) is called an $(n-r)$-dimensional linear manifold.

An $(n-r)$-dimensional linear manifold may, thus, be considered either as a translated set of linear combinations of $(n-r)$ linearly independent vectors, or as the intersection of r linearly independent hyperplanes.

The proof of the equivalence of the two definitions following from Theorems 2.3 and 2.5 is dealt with in the theory of linear equations.

In Chapter 3 analogs of Theorems 2.3 and 2.5 were established with regards to systems of linear equalities and inequalities. These propositions make it possible to prove the equivalence of the two different definitions of a convex polyhedral set — the range of the linear form of the linear-programming problem — which is a generalization of the linear manifold.

2-6. We give now another criterion of consistency of systems of linear equations which in some cases proves to be more convenient than Theorem 2.1.

Theorem 2.6. For a system of linear equations (2.1) to be consistent it is necessary and sufficient that the equality

$$(Y, B) = 0 \tag{2.24}$$

be satisfied for any vector $Y = (y_1, y_2, \ldots, y_m)$ *satisfying the homogeneous system*

$$(A_j, Y) = \sum_{i=1}^{m} a_{ij} y_i = 0, \qquad j = 1, 2, \ldots, n. \tag{2.25}$$

The proof proceeds from the following lemma.

Lemma 2.1. Let $X_1, X_2, \ldots, X_k$ *be an arbitrary linearly independent system of vectors. The matrix*

$$M = \begin{Vmatrix} (X_1, X_1) & (X_1, X_2) \ldots (X_1, X_k) \\ (X_2, X_1) & (X_2, X_2) \ldots (X_2, X_k) \\ \ldots & \ldots \\ (X_k, X_1) & (X_k, X_2) \ldots (X_k, X_k) \end{Vmatrix},$$

whose elements are the scalar products of these vectors, is regular.

Proof. Consider a homogeneous system of equations with the matrix of coefficients M:

$$\sum_{j=1}^{k} (X_i, X_j) y_j = 0, \qquad i = 1, 2, \ldots, k. \tag{2.26}$$

Let $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ be an arbitrary solution of system (2.26):

$$\sum_{j=1}^{k} (X_i, X_j) \lambda_j = 0, \qquad i = 1, 2, \ldots, k. \tag{2.27}$$

Using properties 2 and 3 of the scalar product we rewrite equations (2.27) in the form

$$(X_i, R) = 0, \qquad i = 1, 2, \ldots, k, \tag{2.28}$$

where

$$R = \sum_{j=1}^{k} \lambda_j X_j.$$

Multiplying the i-th equation of system (2.28) by λ_i and summing the results, we obtain

$$\sum_{j=1}^{k} \lambda_i (X_i, R) = \left(\sum_{i=1}^{k} \lambda_i X_i, R\right) = (R, R) = 0. \tag{2.29}$$

Equation (2.29) indicates that

$$R = \sum_{j=1}^{k} \lambda_j X_j = 0.$$

But, by assumption, the vectors $X_1, X_2, \ldots, X_k$ are linearly independent. Hence

$$\lambda_1 = \lambda_2 = \ldots = \lambda_k = 0.$$

System (2.26) has thus a unique solution, namely the zero vector and according to Theorem 2.4 M is therefore a regular matrix. This completes the proof.

Proof of Theorem 2.6.

1. **Necessity.** Let system (2.1) be solvable. This means that the vector B can be represented as a linear combination of vectors A_j (the columns

of the matrix A):

$$B=\sum_{j=1}^{n}\alpha_j A_j. \tag{2.30}$$

Consider an arbitrary solution $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_m)$ of system (2.25):

$$(A_j, \Lambda)=0, \qquad j=1, 2, \ldots, n. \tag{2.31}$$

Scalar-multiplying both sides of the vector equality (2.30) by Λ and applying (2.31), we obtain

$$(B, \Lambda)=\sum_{j=1}^{n}\alpha_j (A_j, \Lambda)=0.$$

This proves necessity.

2. **Sufficiency**. This part of the proof is somewhat more complicated than the preceding. According to Theorem 2.2, among the solutions of the homogeneous system (2.25) there are $(m-r)$ linearly independent vectors (here r is the rank of the matrix A). Let these vectors be

$$Y_i=(y_1^{(i)}, \ y_2^{(i)}, \ldots, y_m^{(i)}), \qquad i=1, 2, \ldots, m-r.$$

By assumption,

$$(A_j, Y_i)=0, \qquad i=1, 2, \ldots, m-r; \ \ j=1, 2, \ldots, n. \tag{2.32}$$

Among the vectors A_j constituting the matrix of coefficients A of system (2.1) there are r linearly independent vectors. To be specific, let the linearly independent vectors be

$$A_1, \ A_2, \ \ldots, \ A_r.$$

We shall show that the vectors

$$A_1, \ A_2, \ \ldots, \ A_r, \quad Y_1, \ Y_2, \ \ldots, \ Y_{m-r} \tag{2.33}$$

constitute a linearly independent system. Let

$$\sum_{j=1}^{r}\alpha_j A_j+\sum_{i=1}^{m-r}\beta_i Y_i=0. \tag{2.34}$$

Scalar-multiplying both terms of (2.34) by $A_l\,(l=1, 2, \ldots, r)$ and $Y_j\,(j=1, 2, \ldots, m-r)$ and applying (2.32), we obtain

$$\sum_{j=1}^{r}\alpha_j (A_j, A_l)=0, \qquad l=1, 2, \ldots, r,$$
$$\sum_{i=1}^{m-r}\beta_i (Y_i, Y_j)=0, \qquad j=1, 2, \ldots, m-r.$$

According to Lemma 2.1 the matrices $\|(A_j, A_l)\|_r$ and $\|(Y_i, Y_j)\|_{m-r}$ are regular. Hence, from Theorem 2.4 we have

$$\alpha_1=\alpha_2=\ldots=\alpha_r=\beta_1=\beta_2=\ldots=\beta_{m-r}=0.$$

Thus, the system of vectors (2.33) is linearly independent. The number of vectors in the system is $r+(m-r)=m$. Therefore, any vector in the m-dimensional space can be represented as a linear combination of these vectors. In particular,

$$B=\sum_{j=1}^{r}x_j A_j+\sum_{i=1}^{m-r}y_i Y_i. \tag{2.35}$$

Multiplying both sides of (2.35) by Y_j and applying the conditions of the

theorem, according to which

$$(B, Y_l)=0, \qquad l=1, 2, \ldots, m-r,$$

and relationship (2.32), we obtain

$$\sum_{l=1}^{m-r} y_l (Y_l, Y_j)=0, \qquad j=1, 2, \ldots, m-r. \tag{2.36}$$

By virtue of Lemma 2.1 and Theorem 2.4, relationships (2.36) lead to the equalities

$$y_1=y_2=\ldots=y_{m-r}=0.$$

Therefore, (2.35) may be rewritten in the form

$$B=\sum_{j=1}^{r} x_j A_j,$$

which indicates that system (2.1) is consistent, one of its solutions being the vector $(x_1, x_2, \ldots, x_r, 0, 0, \ldots, 0)$. This completes the proof.

If the rank of matrix A is r, the general solution of system (2.25) is the set of all linear combinations of some $(m-r)$ linearly independent solutions of this system. The conditions of orthogonality (2.24) should, therefore, be checked only for the $(m-r)$ linearly independent vectors which solve (2.25).

According to Theorem 2.4, the solution of the consistent system (1.2) is unique if the columns of matrix A are linearly independent. Condition (2.24) does not impose limitations on vector B only if $r=m$. Consequently, the criterion of solvability of system (2.1) for any free-term vector reduces to linear independence of the rows of matrix A.

Hence, for the solution of system (2.1) to be unique for any vector B it is necessary and sufficient that A be a square regular matrix.

The following simple proposition (Exercise 8) is also noteworthy:

If A is a square matrix, uniqueness of solution of the homogeneous system corresponding to system (2.1) is a necessary and sufficient condition for the consistency of system (2.1) for any vector B.

2-7. Until now we have not actually solved a system of linear equations. One way of solving a system of the form (2.1) follows directly from the preceding theory of linear systems and is as follows.

First determine the ranks of the matrix of coefficients A and the augmented matrix $\tilde{A}$ of the system.

If these ranks are not equal, the system is unsolvable. Otherwise, we can determine the general solution of the system. To compute the rank of a matrix we must find the regular square submatrix of maximum order. Let it be denoted by $\bar{A}$. The equations whose coefficients are not elements of matrix $\bar{A}$ are then omitted. The unknowns whose coefficients are not elements of matrix $\bar{A}$ are transferred to the right-hand side. We thus obtain a system of equations with a square regular matrix $\bar{A}$ and a free-term vector which is the sum of some vector and the linear combination of several vectors. The values of the coefficients in the linear combination are arbitrary.

Next we apply Cramer's rule, which expresses the unknowns corresponding to the columns of matrix $\bar{A}$ in terms of the remaining unknowns of the system. This completes the construction of the general solution of the system.

The reader will no doubt see that if the parameters m and n are large this method of solving systems of linear equations involves computation of numerous high-order determinants which is most time consuming. Thus,

the above method for finding the general solution of a system of linear equations is not practical.

There are many other, simpler methods for solving systems of linear equations with numerical coefficients. We shall deal here with one.

We refer, as before, to system (2.1). The method described below is known as the *complete elimination method*. This method is due to Jordan and Gauss, and is known also as the *Gauss-Jordan reduction method*.

The complete elimination method is based on two elementary transformations of the augmented matrix.

(1) A row of the augmented matrix is multiplied by a nonzero number.

(2) A row multiplied by an arbitrary number is added to one of the rows of the augmented matrix.

It can be easily verified that each elementary transformation (and consequently any sequence of elementary transformations) produces an augmented matrix of some new system which is equivalent to the initial system. Proof is left to the reader (see Exercise 9).

The complete elimination method has a finite number of steps. In the first step we choose any nonzero element among the elements of the matrix of coefficients A (the principal part of the augmented matrix $\tilde{A}$). This element is called the *direction* [pivot] *element* of the given step (transformation). The row and the column containing the direction element of the transformation are, generally, called the *direction* row and column of the given transformation.

Dividing all the elements of the direction row of the augmented matrix $\tilde{A}$ by the direction element, we obtain a new direction row. Further, from each row of the matrix $\tilde{A}$ (except the direction row) we subtract the new direction row multiplied by the element located at the intersection of the direction column and the row being transformed. This completes the first step of the complete elimination method.

Let $\tilde{A}'$ denote the matrix obtained after this transformation from $\tilde{A}$. The matrix $\tilde{A}'$ was produced from the matrix $\tilde{A}$ by one elementary transformation of type (1) and $(m-1)$ elementary transformations of type (2). Hence, the system of equations with the augmented matrix $\tilde{A}$ is equivalent to the initial system (2.1).

Having carried out the transformations we should have zeroes in the direction column, except for the direction element which should be equal to 1.

All the elements in the first n columns of the matrix $\tilde{A}'$ apart from those in the direction row and direction column as determined in the first step will be called the principal part of the matrix $\tilde{A}'$.

The direction element of the second step is chosen from the nonzero elements of the principal part of the matrix $\tilde{A}'$. The second step is carried out precisely as the first. Observe that the second step does not affect the column chosen in the first step as the direction column, since the element at the intersection of this column with the direction row in the second step is zero. In matrix $\tilde{A}''$, obtained after the second step, the direction columns of the first and second steps are unit vectors. The unit component of each of these vectors is in the corresponding direction row. Subsequent steps of the complete elimination method are carried out analogously. Let $\tilde{A}^{(k)}$ be the matrix obtained after k steps. To each step there corresponds a characteristic direction row and direction column. In matrix $\tilde{A}^{(k)}$ there are k rows and k columns which have been used as direction rows and columns

in the preceding steps. Therefore, k columns of the matrix $\tilde{A}^{(k)}$ are unit vectors with units in the corresponding direction rows. All the elements in the first n columns of the matrix $\tilde{A}^{(k)}$ apart from those in the direction rows and columns of the preceding steps constitute the principal part of this matrix. The direction element of the $(k+1)$-th step is chosen from the nonzero elements of the principal part of matrix $\tilde{A}^{(k)}$. Then, matrix $\tilde{A}^{(k)}$ is subjected to the same sequence of elementary transformations as in the first step resulting in the matrix $\tilde{A}^{(k+1)}$ with $(k+1)$ unit columns.

Subsequent steps are carried out as long as direction elements can be chosen. If after the k-th step the principal part of the matrix $\tilde{A}^{(k)}$ does not contain any elements or comprises zeroes only, the solution process is terminated.

This process, obviously, involves no more than $\tau=\min\{m, n\}$ steps, the total number of multiplications and divisions not exceeding $m\tau\frac{2n-\tau+1}{2}$.

2-8. Let the process of solution terminate after the l-th step. The matrix

$$\tilde{A}^{(l)}=(A_1^{(l)}, A_2^{(l)}, \ldots, A_n^{(l)}, B^{(l)})=(A^{(l)}, B^{(l)})$$

is obtained from the matrix $\tilde{A}$ as the result of elementary transformations. Hence, the system of linear equations

$$A^{(l)}X=B^{(l)} \tag{2.37}$$

is equivalent to system (2.1). Since the process of solution is terminated, the main part of the matrix $\tilde{A}^{(l)}$ either does not contain a single element, or consists of zeroes only.

Let us first assume that among the rows of matrix $\tilde{A}^{(l)}$ there are such which have not served as direction rows in any of the steps. Consider one of these rows, say, the i_0-th row. It is easy to verify that $a_{i_0j}^{(l)}=0$ for $j=1, 2, \ldots, n$. Indeed, if j is the number of the column which served as a direction column in one of the steps, $a_{i_0j}^{(l)}=0$ because the only nonzero element of the j-th column is in the corresponding direction row. If, however, the j-th column was never chosen as a direction column, the element $a_{i_0j}^{(l)}$ is in the main part of the matrix $\tilde{A}^{(l)}$ which, by assumption, contains only zero elements.

Thus, the i_0-th equation of system (2.37) has the form

$$0x_1+0x_2+\ldots+0x_n=b_{i_0}^{(l)}. \tag{2.38}$$

1. If $b_{i_0}^{(l)}\neq 0$, equation (2.38) is contradictory. Therefore, system (2.1), being equivalent to an inconsistent system (2.37), has no solutions.

2. If $b_{i_0}^{(l)}=0$, the i_0-th equation of system (2.37), being an identity $(0=0)$, imposes no restrictions on the unknowns and as such may be omitted.

Examining all the rows of the matrix $\tilde{A}^{(l)}$ which were not used as direction rows we either establish unsolvability fo the system, or else omit all these rows.

Suppose that case (a), which points to unsolvability of system (2.1), did not arise. Then system (2.37) is reduced to those equations which correspond to the direction rows of the matrix $\tilde{A}^{(l)}$.

System (2.37) is thus reduced to precisely l equations (as the number of direction rows). To be specific let us assume these to be the first l equations. We rewrite the resulting system of equations in the form

$$\sum_{j=1}^{n} a_{ij}^{(l)}x_j=b_i^{(l)}, \qquad i=1, 2, \ldots, l. \tag{2.39}$$

System (2.39) is, obviously, equivalent to system (2.37) and hence to the initial system (2.1). In particular, l may be equal to m. This means that all the rows have been chosen as direction rows in the process of solution and, in this case, there is no need to omit any of equations (2.37). The initial system is, thus, either inconsistent, or else reduces to system (2.39) equivalent to (2.1). We shall show that system (2.39) is consistent and that its general solution can be determined without any calculations.

Without loss of generality, take the i-th direction row ($i=1, 2, \ldots, l$) to correspond to the i-th direction column (this can always be achieved by suitably renumbering the unknowns of system (2.39)). Then, following the complete reduction method described above

$$a_{ij}^{(l)} = \begin{cases} 0, & i \neq j, \\ 1, & i = j, \end{cases} \qquad (j=1, 2, \ldots, l).$$

System (2.39), thus, may be rewritten in the form

$$x_i = b_i^{(l)} - \sum_{j=l+1}^{n} a_{ij}^{(l)} x_j, \qquad i=1, 2, \ldots, l. \tag{2.40}$$

We assume that

$$X_0 = (b_1^{(l)}, b_2^{(l)}, \ldots, b_l^{(l)}, 0, 0, \ldots, 0),$$
$$X_s = (\underbrace{-a_{1s}^{(l)}, -a_{2s}^{(l)}, \ldots, -a_{ls}^{(l)}, 0, 0, \ldots, 0,}_{s} 1, 0, \ldots, 0),$$
$$s = l+1, \ldots, n.$$

X_0 obviously solves system (2.40), and the vectors $X_s (s=l+1, \ldots, n)$ satisfy the homogeneous system of equations corresponding to system (2.40).

The rank of the matrix of coefficients of system (2.40) is l (the coefficients of the first l unknowns constitute a unit matrix of order l). The vectors $X_s (s=l+1, \ldots, n)$ are linearly independent (their last $n-l$ components constitute a unit matrix of order $n-l$). Therefore, according to Theorem 2.3, the general solution of system (2.40) has the form

$$X_0 + \sum_{s=l+1}^{n} \alpha_s X_s, \tag{2.41}$$

where $\alpha_{l+1}, \alpha_{l+2}, \ldots, \alpha_n$ are arbitrary coefficients.

System (2.40), however, is equivalent to system (2.1). Hence, (2.41) is the general solution of system (2.1).

Each step in this method results in the elimination of some unknown (corresponding to the direction column) from all the equations of the system except the one corresponding to the direction row of the particular step. This explains the name of the method — complete elimination (reduction) method.

Systems of linear equations are also often solved by the Gauss reduction method which is similar to the complete elimination method. The difference is that in the Gauss reduction method each step results in the elimination of some unknown only from those equations which do not correspond to the direction rows of the preceding steps. Thus, the number of the transformed rows in the k-th step of the Gauss reduction method is $n-k$, whereas in the complete elimination method all the rows are transformed in each step.

Comparing the amount of work involved in each method, we see that the complete elimination method involves more operations than the reduction

method. However, the complete elimination method is of special significance in linear programming since it provides the algorithmic basis for all the finite linear-programming methods.

§ 3. Convex sets

3-1. In 1-2 we defined the n-dimensional Euclidean space E_n. Here we deal with various sets of points (vectors) in the space E_n. If all the points of set G_1 belong to set G_2 this is denoted by $G_1 \subset G_2$ (G_1 is contained in G_2). If the set G_1 contains only one point X, and this point belongs to set G_2, this is denoted thus:

$$X \in G_2.$$

In particular, the notation $G \subset E_n$ indicates that the set G consists of points of space E_n. At the end of the first section we introduced some important sets of the space E_n, namely the r-dimensional subspace, the r-dimensional linear manifold $(0 \leqslant r \leqslant n)$. Particular cases of a linear manifold are the hyperplane $(r = n-1)$ and the line $(r = 1)$.

Consider any hyperplane with the equation

$$(\Lambda, X) = \sum_{i=1}^{n} \lambda_i x_i = c. \tag{3.1}$$

The hyperplane (3.1) divides the Euclidean space into two half-spaces denoted by

$$(\Lambda, X) \leqslant c, \quad (\Lambda, X) \geqslant c. \tag{3.2}$$

Any line in the space E_n has the equation

$$X = A + Bt, \quad -\infty < t < \infty, \tag{3.3}$$

where A and B are some vectors in E_n. Vector B is generally called the direction vector of the given line. If the parameter t in equation (3.3) is bounded above or below by a finite number, the set of points is called a ray (or a half-line) with direction vector B. If the parameter t is bounded above and below by a finite number, equation (3.3) defines a segment.

Any point of a segment specified by the equation

$$X = A + Bt, \quad \alpha \leqslant t \leqslant \beta,$$

can be represented as a linear combination of the end points:

$$A + B\alpha, \quad A + B\beta.$$

Indeed

$$A + Bt = \mu_1 (A + B\alpha) + \mu_2 (A + B\beta), \tag{3.4}$$

$$\mu_1 = \frac{\beta - t}{\beta - \alpha}, \quad \mu_2 = \frac{t - \alpha}{\beta - \alpha}. \tag{3.5}$$

Note that

$$\mu_1 + \mu_2 = 1, \quad \mu_1 \geqslant 0, \quad \mu_2 \geqslant 0. \tag{3.6}$$

Conversely, it can easily be shown that for any μ_1 and μ_2 satisfying (3.6), there exists a number t between α and β such that equalities (3.5) are satisfied and, consequently, (3.4) applies.

A segment whose end points are A' and A'' is thus a set of points X of the form

$$X=\mu A'+(1-\mu)A'', \quad 0\leq\mu\leq 1.$$

A sphere of radius $\varrho>0$ with its center at the point $A\in E_n$ is defined as the set of all points $X\in E_n$ such that

$$|X-A|<\varrho. \tag{3.7}$$

The sphere (3.7) constitutes a ϱ-neighborhood of the point A. If, together with point X, the set $G\in E_n$ also contains the ε-neighborhood of this point for some $\varepsilon>0$, X is said to be an inside point of the set G. An outside point of the set G has a certain neighborhood which is located outside G.

If any neighborhood of a point X contains points both belonging to G and points not belonging to G, then, by definition, X is a boundary point of the set G. A set containing all its boundary points is said to be closed.

It can easily be shown that any linear manifold (in particular, a hyperplane and a line), a half-space, a ray, and a segment are closed sets (see Exercise 11).

Furthermore, it can easily be verified that the union of a finite number of closed sets $G_1, G_2, \ldots, G_k$ $(\bigcup_{i=1}^{k} G_i)$ and the intersection of any number of closed sets $G_1, G_2, \ldots, G_k, \ldots$ $(\bigcap G_i)$ are also closed sets. Proof is left to the reader (see Exercise 12).

A set G is said to be bounded, if there exists a number c independent of X such that

$$|X|<c$$

for all $X\in G$.

The famous Bolzano-Weierstrass theorem is readily extended to the n-dimensional Eucledian space:

From any bounded sequence of points of space E_n we can choose a convergent subsequence. The proof of this theorem is left to the reader (Exercise 13).

Let $F(X)$ be a function which relates to each point X_0 of the set $G\subset E_n$ a definite real number $F(X_0)$. The set G is called the domain of definition of the function $F(X)$.

The function $F(X)$ is said to be continuous at the point X_0 in its domain of definition if for any $\varepsilon>0$ there exists $\delta>0$ such that for any point X in the domain of definition of $F(X)$ separated from X_0 by less than δ $(|X-X_0|<\delta)$

$$|F(X)-F(X_0)|<\varepsilon.$$

A function defined and continuous at each point of set G is said to be continuous over the set G. If $F(X)$ is a function continuous over a bounded closed set G, it attains its upper and lower bounds in the set, i. e., there exists points $X', X''\in G$, such that

$$F(X')=\sup_{X\in G} F(X)=\max_{X\in G} F(X),$$
$$F(X'')=\inf_{X\in G} F(X)=\min_{X\in G} F(X).$$

The proof of this proposition is based on the Bolzano-Weierstrass theorem and is carried out precisely as for a function of one variable (see Exercise 14). Simple examples of continuous functions are the linear function $F(X)=(C,X)=\sum_{i=1}^{n} c_i x_i$ and the quadratic function $F(X)=X^T\|a_{ij}\|X=\sum_{j=1}^{n}\sum_{i=1}^{n} a_{ij}x_i x_j$.

Each of these functions is continuous on the entire space E_n (see Exercise 15).

3-2. A set $G \subset E_n$ is said to be convex if, together with any two points A and B, it also contains the segment joining these points:

$$\mu A + (1-\mu) B, \quad 0 \leq \mu \leq 1.$$

It can easily be verified that if G is a convex set and points $P_1, P_2, \ldots, P_s$ are contained in G, then for any $\alpha_i \geq 0$, $i = 1, 2, \ldots, s$ satisfying the condition

$$\sum_{i=1}^{s} \alpha_i = 1,$$

the point $\sum_{i=1}^{s} \alpha_i P_i \in G$ (see Exercise 16).

All the sets mentioned in 3-1 are convex sets (see Exercise 17).

It can be shown that the intersection of any number of convex sets is a convex set (see Exercise 18). An important class of convex sets comprises the so-called convex cones, defined as follows.

We shall say that a closed convex set $T \subset E_n$ is a convex cone with apex at point P_0, if for any vector $P \in T$ and for any $\mu \geq 0$ the vector

$$P_0 + \mu (P - P_0) \in T.$$

It can easily be verified that any linear manifold is a convex cone with apex at any point of the manifold (see Exercise 19). Another example of a convex cone is the following. Let D be an arbitrary closed convex bounded set, and P_0 some point outside D. The set of all rays issuing from P_0 and crossing D is a convex cone (see Exercise 20).

We shall now state and prove one fundamental proposition relating to convex sets. This proposition is the basis of a number of important facts in the theory of mathematical programming.

We first define: we shall say that a hyperplane

$$(\Lambda, X) = c$$

separates the sets G_1 and G_2 if $(\Lambda, X) \leq c$ for all $X \in G_1$ and $(\Lambda, X) \geq c$ for all $X \in G_2$. In other words, a hyperplane separates the sets G_1 and G_2 if and only if G_1 lies in one of the half-spaces generated by the hyperplane and G_2 in the other.

If all the inequalities in the definition of the separating hyperplane are strict inequalities, we say that the given hyperplane strictly separates the sets G_1 and G_2.

A hyperplane strictly separating the sets G_1 and G_2 has, thus, the following properties: all the points of set G_1 are inside points of one of the half-spaces generated by the hyperplane, and all the points of set G_2 are inside points of the other half-space.

If a set G_1 lies inside one of the half-spaces generated by hyperplane Π, and G_2 in the other half-space, we say that the hyperplane Π strictly isolates the set G_1 from the set G_2.

Theorem 3.1. (Hyperplane separation theorem.) Let G_1 and G_2 be any two closed convex sets having no points in common, and let at least one of the two be bounded. With these assumptions there exists a hyperplane strictly separating the sets G_1 and G_2.

Proof. To be specific, let G_1 be the bounded set.

Let

$$\sigma = \inf |P_1 - P_2|,$$

where the lower bound is taken over all the possible points $P_1 \in G_1$ and $P_2 \in G_2$. By definition of the lower bound, there exist sequences $\{P_1^{(k)}\}$ and $\{P_2^{(k)}\}$ $(P_1^{(k)} \in G_1$, $P_2^{(k)} \in G_2)$ such that

$$\lim_{k \to \infty} |P_1^{(k)} - P_2^{(k)}| = \sigma. \tag{3.8}$$

Since G_1 is a bounded set, sequence $\{P_1^{(k)}\}$ is bounded. Sequence $\{P_2^{(k)}\}$ is also bounded, since

$$|P_2^{(k)}| = |(P_2^{(k)} - P_1^{(k)}) + P_1^{(k)}| \leqslant |P_1^{(k)} - P_2^{(k)}| + |P_1^{(k)}|.$$

Applying the Bolzano-Weierstrass theorem, we choose from the sequence $\{P_1^{(k)}\}$ and $\{P_2^{(k)}\}$ two subsequences converging to the points P_1^* and P_2^*, respectively. Since the sets are closed, $P_1^* \in G_1$, $P_2^* \in G_2$. Thus, there exists a sequence of superscripts k_i, $i = 1, 2, \ldots$, such that

$$\lim_{i \to \infty} |P_1^{(k_i)} - P_1^*| = 0; \quad \lim_{i \to \infty} |P_2^{(k_i)} - P_2^*| = 0.$$

Passing to the limit in $|P_1^{(k_i)} - P_2^{(k_i)}|$ as $i \to \infty$ and applying (3.8), we obtain

$$|P_1^* - P_2^*| = \sigma = \min_{\substack{P_1 \in G_1 \\ P_2 \in G_2}} |P_1 - P_2|. \tag{3.9}$$

By assumption, the sets G_1 and G_2 do not intersect, therefore $\sigma > 0$.

Let

$$R_\mu = \mu P_1^* + (1 - \mu) P_2^*,$$

where $0 < \mu < 1$. Let Π_μ be the hyperplane defined by the equation

$$(\Lambda, X) = (\Lambda, R_\mu), \tag{3.10}$$

where $\Lambda = P_1^* - P_2^*$. Substituting we observe that Π_μ contains the point R_μ. We shall show that the hyperplane Π_μ strictly separates the sets G_1 and G_2. The proof is carried out by reductio ad absurdum.

Let $Q \in G_1$ and also

$$(\Lambda, Q) \leqslant (\Lambda, R_\mu). \tag{3.11}$$

Since P_1^* and Q belong to the convex set G_1, any point $Q_\varepsilon = (1 - \varepsilon) P_1^* + \varepsilon Q$, for $0 \leqslant \varepsilon \leqslant 1$, also belongs to G_1.

We calculate the square of the distance between the points $Q_\varepsilon \in G_1$ and $P_2^* \in G_2$:

$$|Q_\varepsilon - P_2^*|^2 = |(1 - \varepsilon) P_1^* + \varepsilon Q - P_2^*|^2 = |\Lambda + \varepsilon (Q - P_1^*)|^2 =$$
$$= \varepsilon^2 |Q - P_1^*|^2 + 2\varepsilon (\Lambda, Q - P_1^*) + |\Lambda|^2.$$

We shall now show that

$$(\Lambda, Q - P_1^*) < 0. \tag{3.12}$$

Indeed,

$$(\Lambda, R_\mu) = (\Lambda, \mu P_1^* + (1 - \mu) P_2^*) =$$
$$= (\Lambda, (\mu - 1)(P_1^* - P_2^*) + P_1^*) =$$
$$= (\Lambda, (\mu - 1)\Lambda + P_1^*) = (\mu - 1)|\Lambda|^2 + (\Lambda, P_1^*),$$

but $|\Lambda|^2 = \sigma^2 > 0$, $\mu \leqslant 1$. Hence,

$$(\Lambda, R_\mu) < (\Lambda, P_1^*).$$

Comparing this result with (3.11), we obtain

$$(\Lambda,\ P_1^*) > (\Lambda,\ Q),$$

which is equivalent to (3.12).

Let $(\Lambda,\ Q-P_1^*)=\alpha\ (\alpha<0)$, $|Q-P_1^*|=\beta$. Then

$$|Q_\varepsilon - P_2^*| = \beta^2\varepsilon^2 + 2\varepsilon\alpha + \sigma^2.$$

Taking $\theta>0$ such that

$$\frac{-\alpha}{1+\beta^2}\theta < 1,\ \theta < 2,$$

we assume

$$\varepsilon_0 = -\frac{\alpha}{1+\beta^2}\theta.$$

The point $Q_{\varepsilon_0}\in G_1$, since $0<\varepsilon_0<1$. On the other hand

$$|Q_{\varepsilon_0} - P_2^*| = \sigma^2 + 2\varepsilon_0\alpha + \varepsilon_0^2\beta^2 \leqslant \sigma^2 - \frac{\alpha^2}{1+\beta^2}\theta(2-\theta) < \sigma^2.$$

Thus, having assumed (3.11), we obtain a contradiction to (3.9). Hence,

$$(\Lambda,\ Q) > (\Lambda,\ R_\mu)$$

for any $Q\in G_1$. Analogously we show that

$$(\Lambda,\ Q) < (\Lambda,\ R_\mu)$$

for all $Q\in G_2$.

The hyperplane

$$(\Lambda,\ X) = c,$$

where $\Lambda = P_1^* - P_2^*$, $c=(\Lambda,\ R_\mu)$, $R_\mu = \mu P_1^* + (1-\mu)P_2^*$, $0<\mu<1$, thus strictly separates the sets G_1 and G_2. This completes the proof.

If we omit at least one of the assumptions imposed on sets G_1 and G_2 in Theorem 3.1, the proposition no longer applies (see Exercise 21).

Sometimes it is not necessary that the hyperplane strictly separates two sets and it suffices that the hyperplane only separates these sets. In these cases the following proposition is useful.

Theorem 3.2. *If G_1 and G_2 are any two nonintersecting convex sets, there exists a hyperplane separating G_1 and G_2.*

Theorem 3.2 is not used in what follows and therefore the proof is not given here. The interested reader is referred to Exercise 23.

3-3. As a corollary of the theorem of separating hyperplanes, we have two propositions which are used in several chapters.

We shall say that the hyperplane Π defined by the equation

$$(\Lambda,\ X) = c$$

is the support hyperplane of the set G (the support of G) at the point $P_0\in G$ if

(a) $(\Lambda,\ P_0) = c$ (Π contains the point P_0),

(b) $(\Lambda,\ P)\leqslant c$ for all $P\in G$

or

$$(\Lambda,\ P)\geqslant c \text{ for } P\in G$$

(the set G lies in one of the half-spaces generated by Π).

Corollary 3.1. *(Theorem of the support hyperplane.) If P_0 is a boundary point of a closed convex set G, there exists a support hyperplane of G at the point P_0.*

Proof. Since P_0 is a boundary point of the set G, there exists a sequence $\{P_k\}$ of points outside G which converges to P_0. Consider the convex sets G_1 and G_2, the former being the point P_k, and the latter coinciding with G. According to Theorem 3.1 there exists a hyperplane Π_k defined by

$$(\Lambda_k, X) = c_k, \tag{3.13}$$

such that

$$(\Lambda_k, P_k) > c_k, \tag{3.14}$$
$$(\Lambda_k, P) < c_k, \qquad P \in G. \tag{3.15}$$

Without loss of generality we may take

$$|\Lambda_k| \leqslant 1, \quad |c_k| \leqslant 1.$$

Applying the Bolzano-Weierstrass theorem, we choose from the vector sequence $\{\Lambda_k\}$ and the number sequence $\{c_k\}$ two subsequences converging to Λ and c, respectively. We shall show that the hyperplane

$$(\Lambda, X) = c$$

is the required support hyperplane.

Indeed, passing in (3.15) to the limit over the subsequence of the subscripts k for a fixed $P \in G$, we obtain

$$(\Lambda, P) \leqslant c, \qquad P \in G. \tag{3.16}$$

In particular,

$$(\Lambda, P_0) \leqslant c. \tag{3.17}$$

On the other hand, passing to the same limit in (3.14), we obtain the inequality

$$(\Lambda, P_0) \geqslant c. \tag{3.18}$$

Comparing (3.17) and (3.18) we obtain

$$(\Lambda, P_0) = c. \tag{3.19}$$

Relationships (3.16) and (3.19) show that the hyperplane $(\Lambda, X) = c$ is the support hyperplane of G at the point P_0. This completes the proof.

The conditions of the theorem of the support hyperplane can be weaker: it is not necessary to assume that the set G is convex (see Exercise 22).

We define the distance $\varrho(P, G)$ between the point P and the set G as

$$\varrho(P, G) = \inf_{Q \in G} |P - Q|.$$

Let G_1 and G_2 be closed disjoint sets, at least one of which is bounded. A point $P_2^* \in G_2$ is said to be the point of G_2 closest to the set G_1 if

$$\varrho(P_2^*, G_1) = \inf_{P_2 \in G_2} \varrho(P_2, G_1).$$

The existence of this point was established in the proof of Theorem 3.1.

Corollary 3.2. Let G_1 and G_2 be sets satisfying the conditions of Theorem 3.1 and P_2^* be the point of G_2 closest to G_1. There exists a hyperplane Π defined by the equation

$$(\Lambda, X) = c,$$

such that

a) $(\Lambda, P) \leqslant c$ *for* $P \in G_2$,

b) $(\Lambda, P_2^*) = c$,

c) $(\Lambda, P) > c$ *for* $P \in G_1$

(Hyperplane Π is the support of G_2 at the point P_2^* and strictly isolates the set G_1 from the set G_2.)

Proof. We introduce a hyperplane

$$(\Lambda, X) = (\Lambda, R_\mu),$$

where

$$\Lambda = (P_1^* - P_2^*), \quad R_\mu = \mu P_1^* + (1-\mu) P_2^*.$$

Here P_1^* is the point of G_1 closest to G_2.

In the proof of Theorem 3.1 we established that for $0 < \mu < 1$

$$\min_{P_1 \in G_1} (\Lambda, P_1) > (\Lambda, R_\mu), \tag{3.20}$$

$$\max_{P_2 \in G_2} (\Lambda, P_2) < (\Lambda, R_\mu). \tag{3.21}$$

Letting the parameter μ in (3.21) approach zero, we obtain

$$\max_{P_2 \in G_2} (\Lambda, P_2) \leqslant (\Lambda, P_2^*). \tag{3.22}$$

We observe that $(\Lambda, R_\mu) = \mu |\Lambda|^2 + |\Lambda, P_2^*|$ does not increase as μ decreases. Therefore, when the parameter μ in (3.20) approaches zero, the inequality remains strict:

$$\min_{P_1 \in G_1} (\Lambda, P_1) > (\Lambda, P_2^*). \tag{3.23}$$

It follows from (3.22) and (3.23) that the hyperplane Π

$$(\Lambda, X) = (\Lambda, P_2^*)$$

has the properties required and is, thus, the desired hyperplane.

3-4. Consider a three-dimensional space. Among the convex sets in this space we have one-dimensional (segment), two-dimensional (circle), and three-dimensional (sphere) figures. A characteristic property of a three-dimensional convex figure is that there is no plane which can contain it. A two-dimensional convex set can lie in some plane; however, no line contains this set. A one-dimensional convex set always belongs to some line.

These considerations lead to the following definition of a convex set in an n-dimensional space.

We shall say that a convex set has ϱ dimensions if it is contained in some ϱ-dimensional linear manifold and cannot belong to a linear manifold of dimension less than ϱ.

Recalling the two different definitions of a linear manifold, the dimensionality of a convex set can be defined in two equivalent forms:

1. The dimensionality of a convex set $G \subset E_n$ is $\varrho = n - r$, where r is the maximum number of linearly independent hyperplanes whose intersection contains G.

2. The dimensionality of a convex set G is equal to the minimum number ϱ of linearly independent vectors $X_1, X_2, \ldots, X_\varrho$ such that G is contained in the set of all points

$$X = X_0 + \sum_{i=1}^{\varrho} \alpha_i X_i,$$

where $X_0 \in G$, and $\alpha_1, \alpha_2, \ldots, \alpha_\varrho$ are arbitrary numbers.

We now prove a useful proposition which elucidates the concept of dimensionality of a convex set.

Theorem 3.3. *A convex set G has a dimensionality $\varrho \geq 1$ if and only if*

(a) G is contained in some ϱ-dimensional linear manifold R;

(b) there exist a point $P \in G$ and a number $\varepsilon > 0$ such that the ε-neighborhood of the point P the set G and the manifold R coincide.

Proof. 1. **Necessity.** Let G be ϱ-dimensional and let R be a ϱ-dimensional manifold containing G. Consider any point $X_0 \in G$. Choose the maximum number of linearly independent vectors $X_1, X_2, \ldots, X_s$ among the vectors of the form $X - X_0$, where $X \in G$. Their number s is, obviously, equal to ϱ.

Indeed, the inequality $s > \varrho$ contradicts the fact that the set $X - X_0$, $X \in G$ is contained in the ϱ-dimensional subspace $X - X_0$, $X \in R$.

On the other hand, the set G belongs to the manifold

$$X_0 + \sum_{i=1}^{s} \alpha_i X_i,$$

so that $s \geq \varrho$.

Hence $s = \varrho$.

By virtue of the convexity of the set G

$$\alpha_0 X_0 + \sum_{i=1}^{\varrho} \alpha_i (X_i + X_0) \in G$$

for any $\alpha_i \geq 0$, $i = 0, 1, \ldots, \varrho$ whose sum is equal to unity. Hence,

$$X_0 + \sum_{i=1}^{\varrho} \alpha_i X_i \in G, \tag{3.24}$$

if

$$\alpha_i \geq 0, \quad i = 1, 2, \ldots, \varrho, \quad \sum_{i=1}^{\varrho} \alpha_i \leq 1.$$

Let

$$X_0' = X_0 + \frac{1}{2\varrho} \sum_{i=1}^{\varrho} X_i \qquad (X_0' \in G).$$

From (3.24)

$$X_0' \pm \frac{1}{2\varrho} X_i \in G, \qquad i = 1, 2, \ldots, \varrho.$$

Hence,

$$X_0' + \sum_{i=1}^{\varrho} \alpha_i^+ X_i - \sum_{i=1}^{\varrho} \alpha_i^- X_i \in G,$$

if

$$\alpha_i^+ \geq 0, \quad \alpha_i^- \geq 0, \quad i = 1, 2, \ldots, \varrho, \quad \sum_{i=1}^{\varrho} \alpha_i^+ + \sum_{i=1}^{\varrho} \alpha_i^- \leq \frac{1}{2\varrho},$$

or, equivalently,

$$X_0' + \sum_{i=1}^{\varrho} \alpha_i X_i \in G, \tag{3.25}$$

if

$$\sum_{i=1}^{\varrho} |\alpha_i| \leq \frac{1}{2\varrho}.$$

By assumption, the set $X - X_0'$, $X \in G$ is contained in the subspace $X - X_0'$, $X \in R$. Hence, the linearly independent vectors $X_1, X_2, \ldots, X_\varrho$ belong to this subspace and the ϱ-dimensional linear manifold coincides with the set of points of the form

$$X_0' + \sum_{i=1}^{\varrho} \alpha_i X_i, \tag{3.26}$$

where $\alpha_1, \alpha_2, \ldots, \alpha_\rho$ are arbitrary numbers. Let

$$\mu = \inf \left| \sum_{i=1}^{\rho} \alpha_i X_i \right| = \min \sqrt{\sum_{i=1}^{\rho} \sum_{j=1}^{\rho} \alpha_i \alpha_j (X_i, X_j)},$$

where the lower bound is taken over all the possible coefficients α_i satisfying

$$\sum_{i=1}^{\rho} |\alpha_i| = 1, \tag{3.27}$$

The function $\left| \sum_{i=1}^{\rho} \alpha_i X_i \right|$ is defined and continuous at any point

$$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\rho) \in E_\rho.$$

The set of all points $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\rho)$ satisfying (3.27) is a closed bounded set. Hence, there exist numbers $\bar{\alpha}_1, \bar{\alpha}_2, \ldots, \bar{\alpha}_\rho$ such that

$$\mu = \left| \sum_{i=1}^{\rho} \bar{\alpha}_i X_i \right|, \quad \sum_{i=1}^{\rho} \bar{\alpha}_i = 1.$$

By definition, $\mu \geqslant 0$. If $\mu = 0$, then

$$\sum_{i=1}^{\rho} \bar{\alpha}_i X_i = 0,$$

and from the linear independence of the vectors $X_1, X_2, \ldots, X_\rho$,

$$\bar{\alpha}_1 = \bar{\alpha}_2 = \ldots = \bar{\alpha}_\rho = 0,$$

which contradicts (3.27). Hence $\mu > 0$.

Let $X \in R$ and

$$|X - X_0'| \leqslant \frac{\mu}{2\varrho}. \tag{3.28}$$

According to (3.26)

$$X - X_0' = \sum_{i=1}^{\rho} \alpha_i X_i.$$

Let

$$\sum_{i=1}^{\rho} |\alpha_i| = \Delta.$$

According to the definition of μ

$$\frac{1}{\Delta} \left| \sum_{i=1}^{\varrho} \alpha_i X_i \right| = \left| \sum_{i=1}^{\varrho} \frac{\alpha_i}{\Delta} X_i \right| \geqslant \mu.$$

Hence, applying (3.28), we have

$$\Delta = \sum_{i=1}^{\varrho} |\alpha_i| \leqslant \frac{1}{\mu} \left| \sum_{i=1}^{\varrho} \alpha_i X_i \right| \leqslant \frac{1}{2\varrho}. \tag{3.29}$$

From (3.29) and (3.25) we have

$$X = X_0' + \sum_{i=1}^{\rho} \alpha_i X_i \in G.$$

Thus, any point X of the ϱ-dimensional linear manifold R satisfying inequality (3.28) belongs to G, and $G \subset R$. Necessity is thus proved $\left(P = X_0', \ \varepsilon = \frac{\mu}{2\varrho}\right)$.

2. **Sufficiency.** Let R be a ϱ-dimensional linear manifold containing G, and let the point $P \in G$ satisfy (b).

The manifold R consists of points of the form

$$P + \sum_{i=1}^{\rho} \alpha_i X_i,$$

where $X_1, X_2, \ldots, X_\rho$ is a linearly independent system of vectors. According to condition (b) of the theorem,

$$P+\varepsilon X_i \in G$$

for $i=1, 2, \ldots, \rho$. Let R' be an arbitrary linear manifold containing G. In this case the points $\varepsilon X_i (i=1, 2, \ldots, \rho)$ belong to the subspace $X-P$, $X \in R'$. Hence the linear manifold R' cannot have less than ρ dimensions.

The dimensionality of R does not exceed the dimensionality of any linear manifold containing G. This means that G is ρ-dimensional. Thus, sufficiency is proved.

From this theorem it follows, in particular, that convex sets $G \in E_n$ are n-dimensional if and only if they contain inside points.

A zero-dimensional convex set is a point (the intersection of n linearly independent hyperplanes).

A one-dimensional convex set lies on some line. Hence, assuming the set is closed, it coincides with either the line, a ray, or a segment (see Exercise 24).

We emphasize that the definition of dimensionality introduced here refers only to convex sets. Application of this definition to an arbitrary set will not be successful since according to this definition a circle (which is obviously a one-dimensional set) is two dimensional.

3-5. A point P of set G is called an extreme point if it cannot be expressed in terms of two other distinct points P_1 and P_2 in G as

$$P=\mu P_1+(1-\mu) P_2,$$

where

$$0<\mu<1.$$

Thus, if P is an extreme point of set G and S is an arbitrary segment in G, P either does not belong to S or is one of its end points.

Examples of extreme points in linear and plane sets are the end points of a segment, the vertices of a polygonal, the points of the boundary of a circle, etc. Convex sets may have a finite (polygonal) or an infinite (circle) number of extreme points. Some convex sets (e.g., a line) do not contain an extreme point at all. We shall see below that such sets should either be unbounded, or open.

We shall say that the set G is a convex hull of the set G_1 if it is the set of points of the form

$$\sum_{i=1}^{N} \mu_i P_i,$$

where $\sum_{i=1}^{N} \mu_i=1$, $\mu_i \geqslant 0 (i=1, 2, \ldots, N)$ and $P_1, P_2, \ldots, P_N$ is an arbitrary finite system of points in G_1.

The vector $\sum_{i=1}^{N} \mu_i P_i$ is called a convex combination of the vectors $P_1, P_2, \ldots, P_N$. It can easily be shown that the convex hull of any set is a convex set (see Exercise 25). On the other hand, any convex set containing G_1 necessarily contains its convex hull G (see Exercise 16).

The convex hull of a set G is, therefore, the smallest convex set containing G. We shall formulate and prove a proposition which elucidates the structure of a convex closed bounded set.

Theorem 3.4. *(Representation theorem.) Let G be a closed bounded convex set and G^* the set of all extreme points of G. In this case G is the convex hull of the set G^*.*

Proof. Proof is by induction on the dimensionality n of the Euclidean space E_n containing G.

If the bounded closed convex set $G \subset E_1$, it is obviously either a point or a segment. In the first case, the sets G and G^* coincide. In the second case G^* comprises the end points of the segment G, and any segment is a convex hull of its end points. We have thus proved the theorem for $n=1$.

Assume that the proposition is true for the space E_{n-1}. We shall show that it also applies for E_n. Let the set $G \subset E_n$ and $P_0=(p_1^{(0)}, p_2^{(0)}, \ldots, p_n^{(0)})$ be a point in G.

1. First let P_0 be a boundary point of the set G. Then, from Corollary 3.1, we may draw through point P_0 a hyperplane Π with the equation

$$(\Lambda, X)=c, \tag{3.30}$$

which would be the support hyperplane of G, i.e., such that $(\Lambda, X) \leqslant c$ for $X \in G$.

Let G_1 be the intersection of G with the hyperplane Π. Obviously, G_1 is a closed bounded convex set; $P_0 \in G$. To be specific we assume that the last component λ_n of the direction vector $\Lambda=(\lambda_1, \lambda_2, \ldots, \lambda_n)$ of the hyperplane Π is not zero.

According to (3.30), for any point $P=(p_1, p_2, \ldots, p_n) \in \Pi$

$$p_n=-\frac{1}{\lambda_n}\sum_{i=1}^{n-1}\lambda_i p_i+\frac{c}{\lambda_n}. \tag{3.31}$$

Consider the set $\overline{G}_1 \subset E_{n-1}$ comprising the points $\overline{P}=(p_1, p_2, \ldots, p_{n-1}) \in E_{n-1}$ such that the corresponding points $P=(p_1, p_2, \ldots, p_{n-1}, p_n)$, where p_n is determined from (3.31), belong to G_1.

Obviously, $\overline{G}_1$ is a bounded closed convex set (G_1 satisfies these conditions); the point $\overline{P}_0=(p_1^{(0)}, p_2^{(0)}, \ldots, p_{n-1}^{(0)}) \in \overline{G}_1$. Hence, by the assumption of induction, there exist extreme points of the set $\overline{G}_1$,

$$\overline{P}_i=(p_1^{(i)}, p_2^{(i)}, \ldots, p_{n-1}^{(i)}), \qquad i=1, 2, \ldots, N,$$

such that

$$\overline{P}_0=\sum_{i=1}^{N}\mu_i\overline{P}_i, \tag{3.32}$$

$$\sum_{i=1}^{N}\mu_i=1, \quad \mu_i \geqslant 0, \quad i=1, 2, \ldots, N.$$

We define the points

$$P_i=(p_1^{(i)}, p_2^{(i)}, \ldots, p_{n-1}^{(i)}, p_n^{(i)}), \quad i=1, 2, \ldots, N,$$

taking

$$p_n^{(i)}=-\sum_{k=1}^{n-1}\frac{\lambda_k}{\lambda_n}p_k^{(i)}+\frac{c}{\lambda_n}. \tag{3.33}$$

$P_i \in G_1$, since by assumption $\overline{P}_i \in \overline{G}_1$ $(i=1, 2, \ldots, N)$. Since the point P_0 satisfies equality (3.31), and representation (3.32) and relationship (3.33) apply, we have

$$\begin{aligned} p_n^{(0)} &= -\sum_{k=1}^{n-1}\frac{\lambda_k}{\lambda_n}p_k^{(0)}+\frac{c}{\lambda_n}=-\sum_{k=1}^{n-1}\frac{\lambda_k}{\lambda_n}\sum_{i=1}^{N}\mu_i p_k^{(i)}+\frac{c}{\lambda_n}= \\ &=\sum_{i=1}^{N}\mu_i\left[-\sum_{k=1}^{n-1}\frac{\lambda_k}{\lambda_n}p_k^{(i)}+\frac{c}{\lambda_n}\right]=\sum_{i=1}^{N}\mu_i p_n^{(i)}. \end{aligned} \tag{3.34}$$

Applying (3.32) and (3.34), we obtain

$$P_0=\sum_{i=1}^{N}\mu_i P_i, \tag{3.35}$$

where

$$\sum_{i=1}^{N}\mu_i=1;\quad \mu_i\geqslant 0,\qquad i=1,\ 2,\ \ldots,\ N.$$

We shall show that all the points P_i in (3.35) are extreme points of the set G. Assuming the contrary for some $i(1\leqslant i\leqslant N)$, we have

$$P_i=\mu Q_1+(1-\mu)Q_2,\qquad 0<\mu<1,\ Q_1\neq Q_2;$$
$$Q_s=(q_1^{(s)},\ q_2^{(s)},\ \ldots,\ q_n^{(s)})\in G,\qquad s=1,\ 2.$$

Since Π is a support hyperplane of the set G, we have

$$(\Lambda,\ Q_s)\leqslant c,\quad s=1,\ 2. \tag{3.36}$$

Further,

$$(\Lambda,\ P_i)=\mu(\Lambda,\ Q_1)+(1-\mu)(\Lambda,\ Q_2)=c.$$

Hence, applying (3.36) and the condition $0<\mu<1$, we obtain

$$(\Lambda,\ Q_1)=(\Lambda,\ Q_2)=c.$$

Therefore $Q_s\in G_1$, $s=1,2$ and consequently

$$\overline{Q}_s=(q_1^{(s)},\ q_2^{(s)},\ \ldots,\ q_{n-1}^{(s)})\in\overline{G}_1,\ s=1,\ 2.$$

Thus,

$$\overline{P}_i=\mu\overline{Q}_1+(1-\mu)\overline{Q}_2,\ 0<\mu<1,$$
$$\overline{Q}_s\in\overline{G}_1,\quad s=1,\ 2.$$

Since $\overline{P}_i$ is an extreme point of the set $\overline{G}_1$, we have $\overline{Q}_1=\overline{Q}_2=P_i$, whence, applying to relationship (3.31), we have $Q_1=Q_2$. This contradiction shows that P_i is an extreme point of the set G.

2. Now let the point P_0 be an inside point of the set G. Consider a line L

$$X=P_0+Rt,\quad -\infty<t<\infty, \tag{3.37}$$

where R is an arbitrary nonzero vector. The intersection of G and L is, obviously, the segment $\overline{L}$ with

$$X=P_0+Rt,\quad t_1\leqslant t\leqslant t_2.$$

Let

$$X_1=P_0+Rt_1,\quad X_2=P_0+Rt_2.$$

Then

$$P_0=\mu X_1+(1-\mu)X_2,\quad 0<\mu<1.$$

Obviously, X_1 and X_2 are boundary points of G. They can, therefore, be represented in the form (3.35):

$$X_s=\sum_{i=1}^{N_s}\mu_i^{(s)}P_i^{(s)},$$

$$\sum_{i=1}^{N_s}\mu_i^{(s)}=1,\quad \mu_i^{(s)}\geqslant 0,\qquad i=1,\ 2,\ \ldots,\ N_s.$$

$P_i^{(s)}$ are extreme points of G, $s=1,2$. Hence,

$$P_0=\sum_{i=1}^{N_1}\mu\mu_i^{(1)}P_i^{(1)}+\sum_{i=1}^{N_2}(1-\mu)\mu_i^{(2)}P_i^{(2)}.$$

Let

$$P_i = \begin{cases} P_i^{(1)}, & i=1, 2, \ldots, N_1, \\ P_{i-N_1}^{(2)}, & i=N_1+1, N_1+2, \ldots, N_1+N_2=N, \end{cases}$$

$$\mu_i = \begin{cases} \mu\mu_i^{(1)}, & i=1, 2, \ldots, N_1, \\ (1-\mu)\mu_{i-N_1}^{(2)}, & i=N_1+1, N_1+2, \ldots, N. \end{cases}$$

Then

$$P_0 = \sum_{i=1}^{N} \mu_i P_i, \tag{3.38}$$

$$\sum_{i=1}^{N} \mu_i = 1, \quad \mu_i \geqslant 0, \quad i=1, 2, \ldots, N.$$

Thus, according to (3.35) and (3.38), an arbitrary point of the set G can be represented as a convex combination of a finite number of extreme points of this set. And, conversely, any convex combination of a finite number of points of a convex set belongs to this set. G is therefore the convex hull of G^*, the set of its extreme points.

Assuming that the theorem for an $(n-1)$-dimensional space is valid, it is true also for E_n and, as has been shown previously, the theorem applies for $n=1$. This completes the proof.

Let G be a convex set. The set G_0 is said to be the skeleton of G if

(a) G is a convex hull of G_0,

(b) any subset of G_0 does not satisfy (a). Figuratively speaking, a skeleton of a set is the smallest set of all those satisfying (a).

We observe that not every convex set has a skeleton. For example, an open interval does not have a skeleton. The representation theorem establishes the existence of a skeleton for any closed bounded convex set. It is found that the skeleton of set G satisfying these requirements coincides with G^*, the set of extreme points of G.

In conclusion, we give one obvious corollary of Theorem 3.4.

Corollary 3.3. Any nonempty bounded closed convex set contains at least one extreme point.

EXERCISES

1. Applying the properties of scalar multiplication, prove the triangle inequality:

$$|A-B| \leqslant |A-C| + |C-B|,$$

where A, B, C are any points in E_n.

2. Prove the equality

$$\|a_{ij}\|_n \|\tilde{a}_{ij}\|_n = \|\tilde{a}_{ij}\|_n \|a_{ij}\|_n = I_n,$$

where the matrix $\|\tilde{a}_{ij}\|_n$ (the inverse of $\|a_{ij}\|_n$) is defined by (1.7).

3. Prove that the inverse matrix for any regular matrix $\|a_{ij}\|_n$ is unique.

4. Verify that a singular matrix does not have an inverse.

5. If a vector can be written as a linear combination of linearly independent vectors, this representation is unique. Prove.

6. Using the criterion of linear independence of a system of vectors, prove that the rank of any matrix is equal to the maximum number of linearly independent rows (columns) of the matrix.

7. Prove the equivalence of the following two definitions of an r-dimensional subspace:

(a) an r-dimensional subspace is the set of all linear combinations of r linearly independent vectors;

(b) an r-dimensional subspace is a set which, together with any two vectors, contains any linear combination of these vectors, the maximum number of linearly independent vectors in the set being r.

8. If A is a square matrix, solvability of the system of equations

$$AX=B$$

for any vector B is a necessary and sufficient condition for uniqueness of solution of the corresponding homogeneous system

$$AX=0.$$

Prove.

9. Verify that the elementary transformations of the augmented matrix of a system of linear equations introduced in 2-7 do not affect the general solution of the system.

10. Applying the complete elimination method determine the general solution of the following systems:

(a)
$$\begin{aligned} 2x_1+3x_2+x_3-2x_4 \qquad &=1,\\ 2x_1+x_2+3x_3+x_4-3x_5&=3,\\ x_1+2x_2+3x_3-3x_4+2x_5&=2; \end{aligned}$$

(b)
$$\begin{aligned} -x_1+3x_2+2x_3-3x_4+2x_5+3x_6&=3,\\ 2x_1-3x_2-x_3+3x_4-2x_5-x_6&=1,\\ 3x_1-3x_2 \qquad +3x_4-2x_5+x_6&=2; \end{aligned}$$

(c)
$$\begin{aligned} x_1+2x_2+3x_3&=1,\\ -2x_1+x_2-x_3&=2,\\ -x_1+3x_2+2x_3&=3. \end{aligned}$$

11. Verify that any linear manifold, space, ray, and segment are closed sets.

12. Prove that the union of a finite number of closed sets and the intersection of any number of closed sets are closed sets.

13. Prove that for any bounded sequence of points in the space E_n there exists a convergent subsequence.

14. A function defined and continuous on a bounded closed set of space E_n attains its upper and lower bounds on this set. Prove.

15. Show that the linear function

$$F(X)=(C,\ X)=\sum_{i=1}^{n} c_i x_i$$

and the quadratic function

$$F(X)=X^T \| a_{ij} \|_n X=\sum_{j=1}^{n}\sum_{i=1}^{n} a_{ij} x_i x_j$$

are continuous at each point of the space E_n.

16. If G is a convex set, then from

$$P_i \in G,\ \alpha_i \geqslant 0,\quad i=1,\ 2,\ \ldots,\ s,\quad \sum_{i=1}^{s} \alpha_i=1$$

it follows that

$$P=\sum_{i=1}^{s} \alpha_i P_i \in G.$$

Prove.

17. Show that a linear manifold, a half-space, a ray, a segment, and a sphere are convex sets.

18. The intersection of any number of convex sets is a convex set. Prove.

19. Verify that any linear manifold may be considered as a cone with its apex at any point of the manifold.

20. Let D be a bounded closed convex set in the space E_n, and let $P_0 \notin D$. Show that the set of all rays issuing from P_0 and crossing D is a convex cone.

21. Show that omission of any of the assumptions of Theorem 3.1 with regard to the sets G_1 and G_2 renders this proposition, generally speaking, invalid.

22. Let G be a convex set in the space E_n, and let $P \in E_n$ not be an inside point of G. Applying Theorem 3.1 and Corollary 3.1, establish the existence of a hyperplane passing through P and containing G in one of its half-spaces.

23. Prove Theorem 3.2. Hint: consider a set G comprising the points

$$X = X_1 - X_2,$$

where $X_i \in G_i$, $i = 1, 2$ and apply the proposition of the preceding exercise to this set at $P = 0$.

24. Show that a one-dimensional closed convex set coincides either with a line, or with a ray, or with a segment.

25. Prove that the convex hull of any set is a convex set.

26. Improve the proof of the representation theorem by showing that any point of a bounded closed convex set is a convex combination of no more than $n+1$ extreme points of this set. Hint: let the line L with equation (3.37) pass through P_0 and some extreme point of the set G.

27. Show that an open interval has no skeleton.

REFERENCES

by numbers referred to in the text

1. ALEKSANDROV, A., A. LUR'E, and Yu. OLEINIK. Primenenie elektronnykh vychislitel'nykh mashin v operativnom planirovanii (Application of Electronic Computers in Operations Research). — Avtomobil'nyi Transport, No. 6. 1960.

2. ACZEL, M. A. and A. H. RUSSEL. New Methods of Solving Linear Programs. —Oper. Res. Quart., 8 (4): 206-219. 1957.

3. BARSOV, A. S. Chto takoe lineinoe programmirovanie (What is Linear Programming). — Moskva, Fizmatgiz. 1959.

4. BEALE, E. M. L. An Algorithm for Solving the Transportation Problem when Shipping Cost over Each Route is Convex. — Nav. Res. Logist. Quart., Vol. 6, No. 1. 1959.

5. BEALE, E. M. L. An Alternative Method for Linear Programming. — Proc. Cambridge Philos. Soc., 50(4): 513-523. 1954.

6. BEALE, E. M. L. Cycling in the Dual Simplex Algorithm. —Nav. Res. Logist. Quart., 2:269-275. 1955.

7. BEALE, E.M.L. On Minimizing a Convex Function Subject to Linear Inequalities. —J. Roy. Stat. Soc., 17: 173-183. 1955.

8. BEALE, E. M. L. Quadratic Programming. — Nav. Res. Logist. Quart., Vol. 6, No. 3. 1959.

9. BROWN, G. W. Iterative Solution of Games by Fictitous Play. — In: Activity Analysis of Production and Allocation, edited by T. C. Koopmans. Cowles Commission Monograph, No. 13 :374-376. Wiley, New York. 1951.

10. BROWN, G. W. and J. von- NEUMANN. Solution of Games by Differential Equations. —In: Contributions to the Theory of Games, Vol. 1, edited by H. W. Kuhn and A. W. Tucker. Annals of Math. Studies, 23: 73-79. Princeton. 1950.

11. BULAVSKII, V. A. Iterativnyi metod resheniya obshchei zadachi lineinogo programmirovaniya (Iterative Method of Solving the General Linear-Programming Problem). — In: "Chislennye metody optimal'nogo planirovaniya", Ekonomika-matematicheskaya seriya Sibirskogo otdeleniya AN SSSR, No. 1, 35-65. 1962.

12. BATCHELOR, J. H. Bibliography on Operations Research, 2nd edition, St. Louis. 1959.

13. WAGNER, H. M. A Comparison of the Original and the Revised Simplex Methods. — Oper. Res., 5 (3): 361-369. 1957.

14. WAGNER, H. M. The Simplex Method for Beginners. — Oper. Res., 6(2): 190-199. 1958.

15. VAJDA, S. Theory of Games and Linear Programming. — In /96/, pp. 11-106.

16. VOLFE, P. The Simplex Method for Quadratic Programming.—Econometrica, Vol. 27. No. 3. 1959.

17. HABR, J. Linear Programming (An Economist's Guide). [Russian translation. 1960.]

18. GASS, S. I. Linear Programming Methods and Applications. —McGraw-Hill, New-York. 1958. [Russian translation. 1961.]

19. GASS, S. I. and T. L. SAATY. The Computational Algorithm for the Parametric Objective Function. — Nav. Res. Logist. Quart., 2: 39-45. 1955.

20. GEL'FAND, I. M. and M. L. TSEITLIN. Printsip nelokal'nogo poiska v sistemakh avtomaticheskoi optimizatsii (The Principle of Nonlocal Retrieval in Automatic-Optimization Systems). — DAN SSSR, Vol. 137, No. 2. 1961.

21. GERCHUK, Ya. P. Problemy optimal'nogo planirovaniya (lineinoe programmirovanie) (Problems of Optimal Planning (Linear Programming)). — Moskva, Ekonomizdat. 1961.

22. GLASAL. An Algorithm for Solving the Transportation Problem. — Matematika, Vol. 2, No. 1. 1958.

23. GOLDMAN, A. J. Resolution and Separability Theorems for Convex Polyhedral Sets. — In /96/, pp. 162-171.

24. GOLDMAN, A. J. and A. W. TUCKER. Convex Polyhedral Cones. — In /96/, pp. 142-161.

25. GOLDMAN, A. J. and A. W. TUCKER. Linear Programming Theory. — In /96/, pp. 172-213.

26. GOL'SHTEIN, E. G. Ob odnom klasse nelineinykh, ekstremal'nykh zadach (On a Certain Class of Nonlinear Extremum Problems). — DAN SSSR, 133(3): 507-510. 1960.

27. GOL'SHTEIN, E. G. Ob odnom beskonechnom analoge zadach lineinogo programmirovaniya i ego prilozheniyakh k nekotorym voprosam teorii priblizhenii (On a Certain Infinite Analog of the Linear-Programming Problems and Its Applications to Some Problems of the Theory of Approximations). — DAN SSSR, 140(1): 23-25. 1961.

28. GOL'SHTEIN, E. G. Zadacha nailuchshego chebyshevskogo priblizheniya v kompleksnoi oblasti s dopolnitel'nymi usloviyami na koeffitsienty approksimiruyushchego polinoma (Chebyshev Best-Approximation Problem in the Complex Domain with Constrained Coefficients of the Approximation Polynomial). — DAN SSSR, 141(2): 274-276. 1961.

29. GOL'SHTEIN, E. G. Ob odnom obobshchenii chebyshevskoi zadachi nailuchshego priblizheniya (On a Certain Generalization of the Chebyshev Best-Approximation Problem). — Uspekhi Matematicheskikh Nauk, 16. 4(100): 208-210. 1961.

30. GOL'SHTEIN, E. G. Ob odnoi obshchei postanovke zadachi nailuchshego priblizheniya (On a Certain General Statement of the Best-Approximation Problem). — DAN SSSR, 144(1): 21-22. 1962.

31. GOL'SHTEIN, E. G. and D. B. YUDIN. Ob odnom klasse zadach planirovaniya narodnogo khozyaistva (On a Certain Class of Problems of National-Economic Planning). — In: Problemy Kibernetiki, No. 5: 165-182. 1961.

32. GOMORY, R. E. Outline of an Algorithm for Integer Solutions to Linear Programs. — Bull. Amer. Math. Soc., 64(5): 275-278. 1958.

33. GOMORY, R. E. An Algorithm for Integer Solutions to Linear Programs. — Princeton-IBM, Mathematics Research Project, Technical Report No. 1. 17 November 1958.

34. GOMORY, R. E. All-Integer Programming Algorithm. — IBM Res. Center Report No. 189. 19 January 1960.

35. GOMORY, R. E. and W. J. BAUMOL. Integer Programming and Pricing. — Econometrica, 28: 521-550. 1960. [Russian translation. 1962.]

36. HOFFMAN, A. J. Cycling in the Simplex Algorithm. — National Bureau of Standards Report No. 2974. 16 December 1953.

37. HOFFMAN, A. F., M. MANNOS, D. SOKOLOWSKY, and N. A. WIEGMANN. Computational Experience in Solving Linear Programs. — J. Soc. Industr. Appl. Math., 1(1): 17-33. 1953.

38. GALE, D. The Basic Theorems of Real Linear Programming and Game Theory. — Nav. Res. Logist. Quart., 3: 193-200. 1956.

39. GALE, D., H. W. KUHN, and A. W. TUCKER. Linear Programming and the Theory of Games. — In: Activity Analysis of Production and Allocation, edited by T. C. Koopmans. Cowles Commission Monograph, No. 13: 317-329. Wiley, New-York. 1951.

40. DANTZIG, G. B. Programming of Independent Activities, Mathematical Model. — Econometrica, 17: 200-211. 1949.

41. DANTZIG, G. B. Maximization of a Linear Function of Variables Subject to Linear Inequalities. — In: Activity Analysis of Production and Allocation, edited by T. C. Koopmans. Cowles Commission Monograph, No. 13: 339-347. Wiley, New-York. 1951.

42. DANTZIG, G. B. Application of the Simplex Method to a Transportation Problem. — In: Activity Analysis of Production and Allocation, edited by T. C. Koopmans. Cowles Commission Monograph, No. 13: 359-373. Wiley, New-York. 1951.

43. DANTZIG, G. B. A Proof of the Equivalence of the Programming Problem and the Game Problem. — In: Activity Analysis of Production and Allocation, edited by T. C. Koopmans. Cowles Commission Monograph, No. 13: 330-335. Wiley, New-York. 1951.

44. DANTZIG, G. B. Upper Bounds, Secondary Constraints, and Block Triangularity in Linear Programming. — Econometrica, 32(2): 174-183. 1955.

45. DANTZIG, G. B. Constructive Proof of the Min-Max Theorem. — Pacific J. Math., 6(1): 25-33. 1956.

46. DANTZIG, G. B. Note on Solving Linear Programs in Integers. — Nav. Res. Logist. Quart., Vol. 4, No. 1. 1959.

47. DANTZIG, G. B. Linear Programming Under Uncertainty. — Manag. Sci., 1: 197-206. 1954-55.

48. DANTZIG, G. B. and P. WOLFE. Decomposition Principle for Linear Programs. — Oper. Res., 8(1): 101-112. 1960.

49. DANTZIG, G. B. and A. ORDEN. A Duality Theorem Based on the Simplex Method. — In: Symposium on Linear Inequalities and Programming, edited by A. Orden and L. Goldstein, (SCOOP Project, 10), pp. 51-55. Directorate of Management Analysis, DCS Comptroller, Headquarters US Air Force Washington, D.C. 1952.

50. DANTZIG, G. B., A. ORDEN, and P. WOLFE. The Generalized Simplex Method for Minimizing a Linear Form under Linear Inequality Restraints. — Pacific J. Math., 5(2): 183-195. 1955.

51. DANTZIG, G. B. and W. ORCHARD-HAYS. The Product Form for the Inverse in the Simplex Method. — Math. Tables Aids Compl, 8: 64-67. 1954.

52. DANTZIG, G. B., L. R. FORD, and D. R. FULKERSON. An Algorithm for Simultaneous Solution of the Primal and the Dual Linear-Programming Problems. — In /96/.

53. DUFFIN, R. J. Infinite Programs. — In /96/.

54. DENNIS, J. B. Mathematical Programming and Electric Circuits. [Russian translation. 1961.]

55. DUMLER, S. A. Lineinoi programmirovanie i ego primenenie na proizvodstve (Linear Programming and Its Application in Industry). — Vestnik Mashinostroeniya, No. 10: 70-74. 1958.

56. DAVIS, C. Linear Programming and Computers, Part I. — Computers and Automation, No. 4: 1-7, 10-15. 1955; Part II. — Computers and Automation, No. 8: 10-16. 1955.

57. ZOUTENDIJK, G. Methods of Feasible Directions. [Russian translation. 1963.]

58. ZUKHOVITSKII, S. I. Nekotorye voprosy teorii chebyshevskikh priblizhenii (Some Problems of the Theory of Chebyshev Approximations). Kiev. 1950.

59. ZUKHOVITSKII, S. I. Algoritm dlya resheniya chebyshevskoi zadachi v sluchae konechnoi sistemy nesovmestnykh lineinykh uravnenii (An Algorithm for Solving the Chebyshev Problem in the Case of a Finite System of Inconsistent Linear Equations). — DAN SSSR, 79(4): 561-564. 1951.

60. ZUKHOVITSKII, S. I. O nailuchshem v smysle P. L. Chebysheva priblizhenii konechnoi sistemy nesovmestnykh lineinykh uravnenii (On the Best, in the Sense of P. L. Chebyshev, Approximation of a Finite System of Inconsistent Linear Equations). — Matematicheskii sbornik, 33(2): 327-342. 1953.

61. KANTOROVICH, L. V. Matematicheskie metody v organizatsii i planirovanii proizvodstva (Mathematical Methods in Industrial Organization and Planning), 67 pp. — Leningradskii Gosudarstvennyi Universitet. 1939.

62. KANTOROVICH, L. V. Ob odnom effektivnom metode resheniya nekotorykh klassov ekstremal'nykh problem (On a Certain Effective Method of Solving Some Classes of Extremum Problems). — DAN SSSR, 28(3): 212-215. 1940.

63. KANTOROVICH, L. V. O peremeshchenii mass (On the Displacement of Masses). — DAN SSSR, 37(7-8): 227-229. 1942.

2154

64. KANTOROVICH, L. V. O metodakh analiza nekotorykh ekstremal'nykh planovo-proizvodstennykh zadach (On Methods of Analysis of Some Extremum Industrial-Planning Problems). — DAN SSSR, 115(3): 441-444. 1957.

65. KANTOROVICH, L. V. Ekonomicheskii raschet nailuchshego ispol'zovaniya resursov (An Economical Computation of Optimum Exploitation of Resources). — Moskva, Izd. AN SSSR, 1959.

66. KANTOROVICH, L. V. and M. K. GAVURIN. Primenenie matematicheskikh metodov v voprosakh analiza gruzopotokov (Application of Mathematical Methods in Analysis of Freight Flow). — In: Sbornik "Problemy povysheniya effektivnosti raboty transporta", pp. 110-138, Izd. AN SSSR, 1949.

67. KANTOROVICH, L. V. and V. A. ZALGALLER. Raschet ratsional'nogo raskroya promyshlennykh materialov (A Computation of Effective Allocation of Industrial Supplies), 198 pp. — Lenizdat. 1951.

68. KANTOROVICH, L. V. and G. Sh. RUBINSHTEIN. Ob odnom funktsional'nom prostranstve i nekotorykh ekstremal'nykh zadachakh (On a Certain Functional Space and Some Extremum Problems). — DAN SSSR, 115(6): 1058-1061. 1957.

69. KREKO, B. Lectures on Linear Programming. — In /97/, pp. 400-430.

70. KUHN, H. W. The Hungarian Method for Solving the Assignment Problem. — Nav. Res. Logist. Quart., 2: 83-97. 1955.

71. KUHN, H. W. Variants of the Hungarian Method for Assignment Problems. — Nav. Res. Logist. Quart., 3(4): 253-258. 1956.

72. KUHN, H. W. and A. W. TUCKER. Nonlinear Programming. — In: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, edited by J. von Neumann, pp. 481-492. Univ. Calif, Berkeley. 1951.

73. LEMKE, G. E. The Dual Method of Solving the Linear Programming Problem. —Nav. Res. Logist. Quart., 1(1): 36-47. 1954.

74. LEONTIEFF, W. W. A Study of the Structure of American Economy. [Russian translation. 1958.]

75. MADANSKY. Inequalities for Stochastic Linear Programming Problems. — Manag. Sci., 6(2): 197-204. 1960.

76. MUNKRES, J. Algorithms for the Assignment and Transportation Problems. — J. Soc. Industr. Appl. Math., 5(7): 32-38. 1957.

77. MANNE, A. S. A Target-Assignment Problem. — J. Oper. Res. Soc. Am., 6(3): 307-466. 1958.

78. MARKOVITZ, H. The Elimination Form of the Inverse and Its Application to Linear Programming. — Manag. Sci., 3(3): 255-269. 1957.

79. MARKOVITZ, H. and A. S. MANNE. On the Solution of Discrete Programming Problems. — Econometrica, 25(1): 84-110. 1957.

80. MARSCHAK, J. and M. R. MICKEY. Optimal Weapon Systems. — Nav. Res. Logist. Quart., Vols. 1 and 2. 1954.

81. MILLS, H. D. Marginal Values of Matrix Games and Linear-Programming Problems. — In /96/, pp. 287-297.

82. MOTZKIN, T. S. New Techniques for Linear Inequalities and Optimization. — Symposium on Linear Inequalities and Programming, pp. 15-27. Washington. 1952.

83. MOTZKIN, T. S., H. RAIFFA, G. U. THOMPSON, and R. M. THRALL. The Double Description Method. — In: Contributions to the Theory of Games, Vol. 2. Ann. of Math. Studies, No. 28: 51-73. Princeton. 1953.

84. VON NEUMANN, J. A Numerical Method to Determine Optimum Strategy. — Nav. Res. Logist. Quart., 1(2): 109-115. 1954.

85. ORCHARD-HAYS, W. Evolution of Linear Programming Computing Techniques. — Manag. Sci., 4(2): 183-190. 1958.

86. OSBORN, H. The Problem of Continuous Programs. — Pacific J. Math., 6(4): 721-731. 1956.

2154

87. RADNER, R. The Linear Team: An Example of Linear Programming under Uncertainty. —Proceedings of the Second Symposium in Linear Programming, Vols. 1 and 2, edited by H. A. Antosiewicz. pp. 381-396. Directorate of Management Analysis, DCS Comptroller, Headquarters US Air Force, Washington, DC. 1955.

88. REINFELD, N. V. and W. R. FOGEL. Mathematical Programming. [Russian translation. 1960.]

89. REMEZ, E. Ya. Obshchie vychislitel'nye metody chebyshevskogo priblizheniya (General Computational Methods of the Chebyshev Approximation). — Kiev, Izd. AN UkrSSR, 1957.

90. RILEY, V. and S. I. GASS. Bibliography on Linear Programming and Related Techniques. Baltimore. 1958.

91. ROBINSON, J. An Iterative Method of Solving a Game. —Ann. Math., 54: 296-301. 1951.

92. ROHDE, F. V. Bibliography on Linear Programming. — Oper. Res., 5(1): 45-62. 1957.

93. ROSEN, J. The Gradient Projection Method for Nonlinear Programming. — J. Soc. Industr. and Appl. Math., 8: 181-217. 1960.

94. RUBINSHTEIN, G. Sh. Zadacha o krainei tochke peresecheniya osi s mnogogrannikom i ee prilozhenie k issledovaniyu konechnoi sistemy lineinykh neravenstv (The Problem of the Extreme Intersection Point of Axis and Polyhedron and Its Application to an Analysis of a Finite System of Linear Inequalities). — DAN SSSR, 100(4): 627-630. 1955.

95. RUBINSHTEIN, G. Sh. Obobshchenie zadachi o krainei tochki peresecheniya osi s vypuklym mnogogrannikom (Generalization of the Problem on Extreme Intersection Point of Axis with Convex Polyhedron). — DAN SSSR, 113(5): 987-990. 1957.

96. KUHN, H. W. and A. W. TUCKER, editors. Linear Inequalities and Related Systems. — Princeton University Press. 1956. [Russian translation. 1959.]

97. NEMCHINOV, V. S., editor. Primenenie matematiki v ekonomicheskikh issledovaniyakh (The Use of Mathematics in Economic Research), Vol. 1. — Moskva, Sotsekgiz. 1959.

98. NEMCHINOV, V. S., editor. Primenenie matematiki v ekonomicheskikh issledovaniyakh (The Use of Mathematics in Economic Research), Vol. 2. — Moskva, Sotsekgiz. 1962.

99. Primenenie matematiki i elektronnoi tekhniki v planirovanii (The Use of Mathematics and Electronics in Planning). Moskva, Izd ekonomicheskoi literatur. 1961.

100. FEDOROVICH M. M., editor. Matematicheskie metody v planirovanii proizvodstva (Mathematical Methods in Industrial Planning). — Moskva, Izd. ekonomicheskoi literatury. 1961.

101. SETUNOV, F. and Yu. CHERNYSHOV. Primenenie metodov lineinogo programmirovaniya dlya resheniya ekonomicheskikh zadach na avtomobil'nom transporte (Application of Linear-Programming Methods to the Solution of Economic Problems of Automobile Transport). — Trud i zarabotnaya plata, No. 11, 1959.

102. SLATER, M. L. Lagrange Multipliers Revisited: A Contribution to Nonlinear Programming. — Cowles Commission Discussion Paper, Math., 403. November, 1950.

103. TUCKER, A. W. Linear and Nonlinear Programming. — Oper. Res., 5(2): 244-257. 1957.

104. TUCKER, A. W. Dual Systems of Homogeneous Linear Relationships. — In /96/, pp. 126-141.

105. TOMPKINS, C. B. Projection Methods in Calculation. —Proceedings of the Second Symposium in Linear Programming, pp. 425-448. Washington. 1955.

106. Trudy Nauchnogo soveshchaniya po primeneniyu matematicheskikh metodov v ekonomicheskikh issledovaniyakh i planirovanii (Proceedings of the Scientific Conference on the Application of Mathematical Methods in Economic Research and Planning) (4–8 April 1960), Vol. 4, Lineinoe programmirovanie (Linear Programming). — Moskva, Izd. AN SSSR. 1961.

107. UZAWA, H. Theorem on Convex Polyhedral Cones. — In /128/, pp. 43-56.

108. UZAWA, H. The Kuhn-Tucker Theorem on Concave Programming. — In /128/, pp. 57-64.

109. UZAWA, H. The Gradient Method for Concave Programming, II, Global Stability in the Strictly Concave Case. — In /128/, pp. 189-197.

110. FADDEEVA, V. N. Vychislitel'nye metody lineinoi algebry (Computational Methods of Linear Algebra). p 240. —Gostekhizdat. 1950.

111. FORD, L. R. and D. R. FULKERSON. A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem. — Canad. J. Math., 9: 210-218. 1957.

112. FORD, L. R. and D. R. FULKERSON. A Primal-Dual Algorithm for the Capacitated Hitchcock Problem. Nav. Res. Logist. Quart., 4(1): 45-54. 1957.

113. FRISCH, R. La résolution des problemes de programme linéaire par la méthode du potential logarithmique. - Cahier sém. économétrie, 4: 7-20, Discuss., 20-23. 1956.

114. KHAVINSON, S. Ya. Ob odnom klasse ekstremal'nykh zadach dlya polinomov (On a Certain Class of Extremum Problems for Polynomials). — DAN SSSR, 130(5): 997-1000. 1960.

115. HILDRETH, C. Process for Quadratic Programming. — Nav. Res. Logist. Quart., Vol. 3. No. 1. 1957.

116. HITCHCOCK, F. L. Distribution of a Product from Several Sources to Numerous Localities. — J. Math. Phys., 20: 224-230. 1941.

117. CHARNES, A. Optimality and Degeneracy in Linear Programming. — Econometrica, 20(2): 160-170. 1952.

118. CHARNES, A., W. W. COOPER, and A. HENDERSON. An Introduction to Linear Programming.— Wiley, New York, 1958. [Russian translation. 1960.]

119. CHARNES, A. and C. E. LEMPKE. The Minimization of Nonlinear Separable Convex Functionals. — Nav. Res. Logist. Quart., 1(4): 301-312. 1954.

120. SHETTY, C. M. A Solution of Transportation Problem with Nonlinear Costs. - Oper. Res., 7: 571-580. 1959.

121. SHILOV, G. E. Vvedenie v teoriyu lineinykh prostrantsv (Introduction to the Theory of Linear Spaces). — Gostekhizdat. 1956.

122 ABLOW, C. M. and G. BRIGHAM. An Analog Solution of Programming Problems. — Oper. Res., 3: 388-394. 1955.

123. EGERVARY, E. Matrixok combinatorius tulajdonsagairol. — Math. Fiz. Lapok, 38: 16-28. 1931.

124. ARROW, K. J. and L. HURWITCZ. Gradient Method for Constrained Maxima. — Oper. Res., 5(2): 258-265. 1957.

125. ARROW, K. J. and L. HURWITCZ. Gradient Method for Concave Programming, I, Local Results. — In /28/, pp. 175-188.

126. ARROW, K. J. and L. HURWITCZ. Gradient Method for Concave Programming, III, Further Global Results and Application to Resource Distribution. — In /128/, pp. 198-216.

127. ARROW, K. J., L. HURWITCZ, and H. UZAWA. Constraint Qualification in Maximization Problems. — Nav. Res. Logist. Quart., 8(2): 175-191. 1961.

128. ARROW, K. J., L. HURWITCZ. and H. UZAWA. Studies in Linear and Nonlinear Programming. — [Collection of papers in Russian translation.]

129. YUDIN, D. B. and E. G. GOL'SHTEIN. Zadachi i metody lineinogo programmirovaniya (Problems and Methods of Linear Programming). — Sovetskoe radio, Moskva. 1961.

130. YUDIN, D. B. and E. G. GOL'SHTEIN. Ob odnom metode kolichestvennogo analiza uproshchennykh ekonomicheskikh modelei (On a Certain Method of Quantitative Analysis of Simplified Economic Models). — In /98/, pp. 136-182.

131. JAKOBS, W. W. Military Applications of Linear Programming. — Proceedings of the Second Symposium in Linear Programming, Vols. 1 and 2, edited by H. A. Antosiewicz, pp. 1-27. Directorate of Management Analysis, DCS Comptroller, Headquarters US Air Force, Washington DC. 1955.

132. YAKOVLEVA, M. A. Zadacha o minimume transportnykh zatrat (Problem of Minimum Transportation Costs). — In /97/, pp. 390-400.